Patient-Specific Modeling of Normal Pressure Hydrocephalus

by

Mikkel Brudvik Sanderud

Thesis

for the degree of

Master of Science

in

Applied Mathematics –
Computational Science

Faculty of Mathematics and Natural Sciences
University of Oslo

December 6, 2012

Acknowledgements

I would like to thank my supervisors Kent-Andre Mardal, Karen-Helene Støverud and Per Kristian Eide for their advice, encouraging words and patience. Perhaps especially that last part. Without your help I would have drowned, and this thesis with me.

Next, I would thank Svein Linge for helping me with a lowpass filter, Victor Haughton for providing me with articles and information regarding hydrocephalus and Anders Johansen for being my interactive FAQ with respect to elasticity and FEniCS.

My gratitude goes out to all my teachers through the years. Without you, this thesis would never have been an option.

Life as a student can be cold and lonely. A big thank you to all of my fellow students, and especially those of you who contributed in some way to this thesis. You have made life in the trenches a joy.

Finally, for your unconditional love and support (and the occasional home-cooked meal), to my parents Tron and Elisabeth and my sister Karoline: Thank you.

Mikkel Brudvik Sanderud
December, 2012

Contents

Contents **i**

1 Introduction **1**

2 Medical Background **3**
2.1 Cerebrospinal Fluid 3
2.2 Brain Anatomy 3
2.3 Hydrocephalus 5

3 Brain Segmentation **7**
3.1 Data 7
3.2 Brain model generation 8
3.3 ITK-Snap 11

4 Data Analysis **15**
4.1 Analysis of the patient-specific data 16
4.2 Filtering 21
4.3 Results 22
4.4 Summary 24

5 The Finite Element Method **25**
5.1 The Finite Element Method 25
5.2 Function spaces 26
5.3 Formulation 27
5.4 Summary 33

6 Linear Elasticity **35**
6.1 Linear Elasticity 35
6.2 Boundary Value Problem 36
6.3 Variational formulation 36
6.4 Discretization 39
6.5 Algorithm 39
6.6 Boundary Conditions 40
6.7 Summary 41

7 Implementation **43**
7.1 FEniCS 43
7.2 Parameters 50
7.3 Cardiac Cycles 52
7.4 Visualising the results with Paraview 54

8 Results **57**
8.1 Stage I 58
8.2 Stage II 58
8.3 Stage III 61
8.4 von Mises stress 66

9 Discussion **81**
9.1 Mesh generation 82
9.2 Data Analysis 82
9.3 Boundary Conditions 83
9.4 Considerations of our model 84
9.5 New fields of investigation 86

Bibliography **87**

Chapter 1

Introduction

In this thesis, we will simulate the medical condition known as hydrocephalus. We will do this by using the Finite Element Method on a simplified and spherical model of the brain, as well as a three-dimensional brain mesh. The mathematical theory of the Finite Element Method will be derived in Chapter 5. In Chapter 6 we describe linear elasticity, the mathematical theory we use to simulate hydrocephalus on the brain mesh.

Hydrocephalus is a medical condition characterized by enlarged ventricles. Hydrocephalus is perhaps most well known as a condition for infants, in which the head is enlarged. However, it also occurs in adults. It is then a form of dementia. Despite the fact that investigations into the causes of hydrocephalus started in the early 20th century [31], there are still cases of hydrocephalus where the cause is unknown. There are several subgroups of hydrocephalus. One of these groups is Normal-pressure Hydrocephalus (NPH). NPH is again divided into two categories, idiopathic NPH, which is when the cause is unknown, and secondary NPH, which is when the cause is known, such as head trauma or tumors. The focus of this study is investigate possible causes of NPH, and describe NPH as a mechanical system.

Previous studies have also simulated hydrocephalus. Kaczmarek et al. [19] used an idealised cyldrical geometry to simulate the steady-state of the hydrocephalic brain. Nagashima et al. [28] used a two-dimensional mesh to analyse hydrocephalus, and is one of the first in the field to apply the Finite Element Method to this kind of problem. Taylor & Miller [35] used a two-dimensional mesh of a brain to reassess elasticity parameters used for brain tissue. Wirth [37] created meshes that shared characteristics with brains during the process of creating a mathematical model of hydrocephalus.

Our study further the previous work in the field, in the sense that we use the three-dimensional mesh of a brain derived from MRI images. This is described in Chapter 3.

The pressure in the brain is pulsatile. This is because as blood is pumped out into the blood vessels, the blood vessels expand. The brain contains several

blood vessels. As the blood vessels expand, the available space is reduced, leading to an increase in the pressure of the cerebrospinal fluid. We are not aware of any studies that apply this pulsatile pressure in their simulations. Previous studies have considered the amplitude of the pulsatile pressure, such as Eide et al. [11] and Matsumoto et al. [26]. We extend these studies by considering the area under the curve, a more robust method to consider the pulsatile pressure through an entire cardiac cycle. We do this in Chapter 4.

In our simulations we find that the elasticity parameters used in the literature are unsuited for our simulations. They give far too great deformations to be realistic.

Chapter 2

Medical Background

The motivation for this study is the medical condition known as hydrocephalus. Hydrocephalus is a latin term, meaning water head. This name is due to the role the cerebrospinal fluid plays in causing hydrocephalus. We will therefore require some knowledge of the condition and the surrounding medical issues. In our this means the brain and the cerebrospinal fluid (CSF) system.

2.1 Cerebrospinal Fluid

The CSF fills the subarachnoid space, ventricular space and cavities and sulci in the brain, as well as the spinal cord. It behaves in a pulsatile manner, based on the heart flow. During the systole the blood vessels expand, making less room for CSF, thereby increasing the pressure. During the contraction of the heart, this pressure is decreased.

The CSF is a clear liquid in and around the brain tissue. Its purpose is to [16]

- Provide a protective layer for the brain.
- Clean out waste products.
- Deliver nutrients and growth factors important for the neural network.
- Allow blood pulsation through the brain.

2.2 Brain Anatomy

Inside of the skull, there are three layers of meninges. The dura mater is the outermost layer, and sticks to the skull. The middle is the arachnoid, and the innermost is the pia mater. The Subarachnoid Space (SAS) is between the arachnoid and the pia mater.

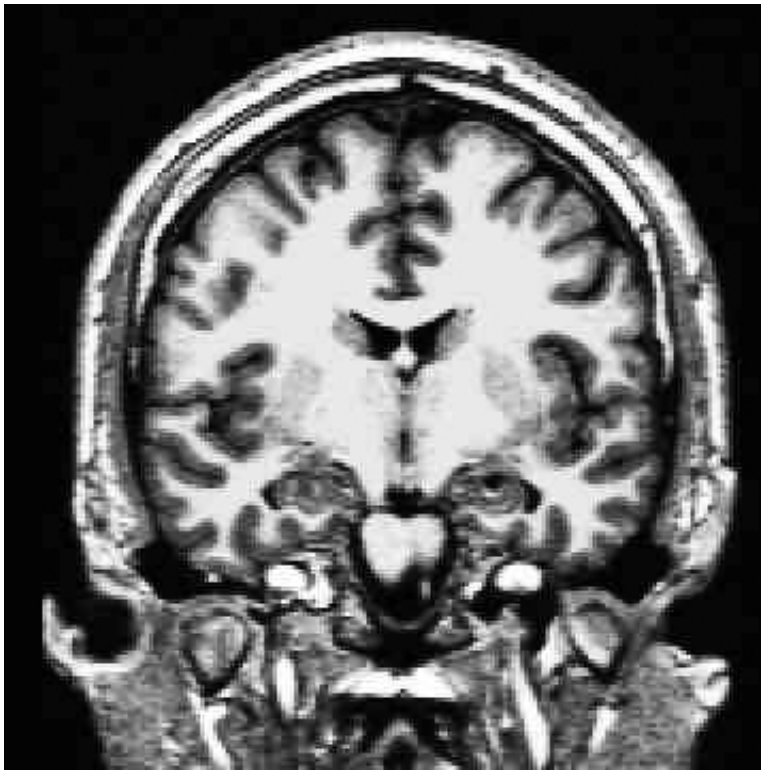


Figure 2.1: Grey and white brain tissue, taken from the webpage *www.med.nyc.edu*.

Under the pia mater we find the brain tissue. The brain tissue is divided into two subgroups, namely grey and white brain matter. The grey matter covers the white matter. A sulcuc is a fissure in the brain, of which there are several. The biggest sulcuc is refered to as the longitudinal cerebral fissure, and divides the brain into two hemispheres.

Inside of the brain we find the Ventricular System (VS). The VS contains four ventricles, and channels between them. The four ventricules are the right and left lateral ventricles, the third ventricle and the fourth ventricle. The CSF is produced in the chloroid plexuses [31], which are located in the ventricles. The VS can be seen in Figure 2.2.

Elasticity

A material is elastic if it is deformed when when pressure it applied to parts of the surface, and it reverts back to its original state when the pressure is removed. We simulate the brain as an elastic material.

Plastic elasticity is when a material is permanently deformed, i.e., that the material does not revert back to its original state when the pressure is let up.

2.3 Hydrocephalus

Normal-pressure hydrocephalus (NPH) is characterized by enlarged cerebral ventricles but normal intracranial pressure (ICP). Symptoms include dementia, loss of balance and bladder control issues. Many patients with NPH demonstrate dramatic improvement when shunted, but it still remains difficult to differentiate between patients that will or will not respond to treatment. A shunt is a drain or a pump designed to channel CSF in the desired direction, to avoid excessive buildup of CSF.

It has been hypothesized that NPH is caused by transmantle static pressure gradients, e.g., Hoff et al. [18], but such pressure gradients have not been observed clinically and hence a basic understanding of the underlying mechanics behind NPH is lacking. A transmantle gradient exists when there is a difference in the pressure in the VS and in the SAS.

Idiopathic NPH (iNPH) is the general term for NPH when the cause is unknown. Secondary NPH (sNPH) is the general term for when the cause is known, e.g., head trauma or a tumor.

Deformation

The main symptom of hydrocephalus is enlarged ventricles in the brain. This can be viewed on MRI scans. There are three different kinds of hydrocephalus. Communicating hydrocephalus is caused by a blockage or obstruction in the brain. Noncommunicating hydrocephalus is caused by a damage in the tissue tasked with absorbing CSF, thus creating an excess of CSF. Normal pressure hydrocephalus (NPH) is characterized by enlarged ventricles *without* any apparent increase in the CSF pressure. One theory is that the enlargement is cause by a transmantle pressure gradient. This has not been proven, however, and investigation on communicating and noncommunicating hydrocephalus has shown that a transmantle gradient is not necessary for enlarged ventricles [34].

It is normally assumed that the deformations occur during four to five days. Afterwards an equilibrium between pressure and brain tissue distribution is obtained. However, in the case of iNPH this is not certain, and might be a process that takes months, or even years.

Previous work

Previous work have also simulated hydrocephalus. Kaczmarek et al. [19] simulated the steady-state of the hydrocephalic brain, by use of an idealised cylindrical model. With their model, they reproduced the steady-state distribution of edema seen in hydrocephalus.

Nagashima et al. [28] was one of the first studies to use simulate hydrocephalus, and they did it on a two-dimensional model.

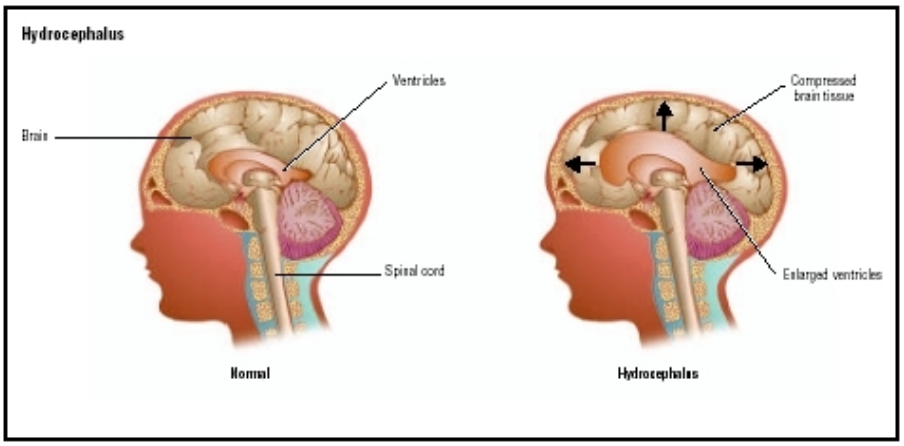


Figure 2.2: The ventricles of a normal person (left) and the enlarged ventricles of a person suffering from hydrocephalus (right). The image is taken from the webpage *www.healthofchildren.com*.

Taylor & Miller [35] reassessed the elasticity parameters for the brain tissue, and used a two-dimensional model. They found lower parametric values than what is used in most of the literature.

Wirth [37] put forward a mathematical model for hydrocephalus, simulated on simplified geometries of the brain.

Chapter 3

Brain Segmentation

Previous work has simulated hydrocephalus on a two-dimensional mesh, e.g., Taylor & Miller [35]. A mesh is a representation of some structure. One natural next step is to simulate hydrocephalus on a three-dimensional mesh, which is what we do in this thesis. The process of obtaining such a mesh is the focus of this chapter.The process of obtaining the mesh can be described in four stages.

- First we obtain MRI images of the brain.
- Then a level set method [30] is used to create a surface of the brain.
- Next a polyhedral surface is created by the marching cubes algorthm [25].
- Finally a volume mesh is created.

This process is illustrated in Figure 3.1.

3.1 Data

Magnetic resonance imaging (MRI) is a scanning technique used in medical imaging. Its basic principle is based on imaging the hydrogen atoms in water and fat [4]. Both water and fat can be found in body tissues. When hydrogen atoms are placed in a magnetic field, their protons are aligned with that field.

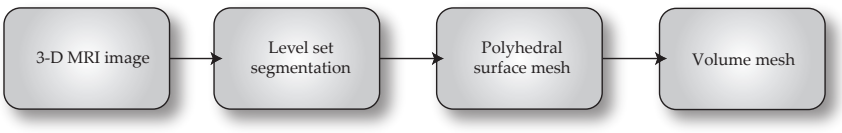


Figure 3.1: The steps in the process of creating a mesh of a brain.

When the magnetic field is turned off, the protons will revert back to their original alignment, i.e., almost exactly half with positive spin and the other half with negative spin. The time it takes the protons to revert back to such an alignment is what is measured during an MRI scan. The reverting is refered to as relaxation. The relaxation times vary among different tissues and materials. The different relaxation times makes it possible to distinguish between various tissues and materials.

3.2 Brain model generation

For our simulations we use a 3D model of a brain. We use the MRI scans of a brain to construct a 3D model of the brain parenchyma. The tool *Vascular Modeling Toolkit* (VMTK) [3] is used to create a segmentation. VMTK is a software designed for generating segmentation and meshes, primarily of blood vessels. Segmentation is the process of identifying and separating different objects and structures in a digital image. In a 3D image this amounts to extracting a volume from a 3D image. The extraction is done by partitioning an image into segments, based on the segments' pixel values. Filtering and manipulation of the raw segmentation is done in *Paraview* [17] and *Meshlab* [1].

Segmentation

Digital Imaging and Communication in Medicine (DICOM) is a file format for medical image data [27]. A DICOM directory consisting of an MRI image of a brain is imported into VMTK. An example is displayed in Figure 3.2. In the figure you can also see the enlarged ventricles typical of a person suffering from iNPH. The grey and white brain tissue were segmented as the same matter. VMTK uses the level set method [30], which is a numerical method for describing curve and surface propagation. The curves and surfaces are described by functions at level zero.

We call the VMTK command `vmtklevelsetsegmentation` after importing the MRI image. The *Colliding Fronts* algorithm is used, as it allows for the most control during the segmentation. A pixel value interval which fits the material we wish to extract is entered. When using the *Colliding Fronts* you place two seeds on the image. From each of these seeds, a front is propagated. These fronts propagate in the previously chosen pixel value interval. The segmentation is the area where the two propagating fronts collide, which is displayed in Figure 3.3.

We do not attempt to segment the whole brain in one single run of an algorithm. Instead, the brain is segmented by repeatedly applying the level set method to one sagittal slice at a time. The segmentation of approximately half the brain is displayed in Figure 3.4. During the segmentation process, different pixel value intervals are used. Different values are necessary because

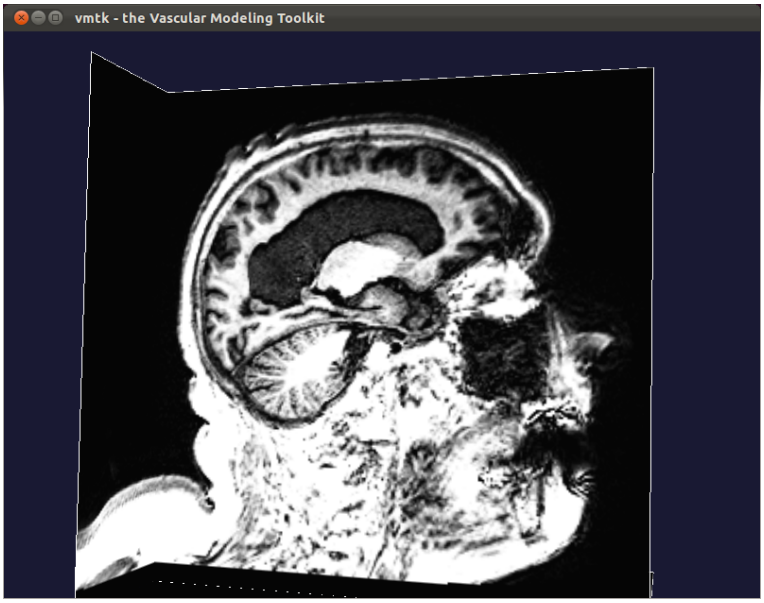


Figure 3.2: Our MRI image viewed by VMTK.

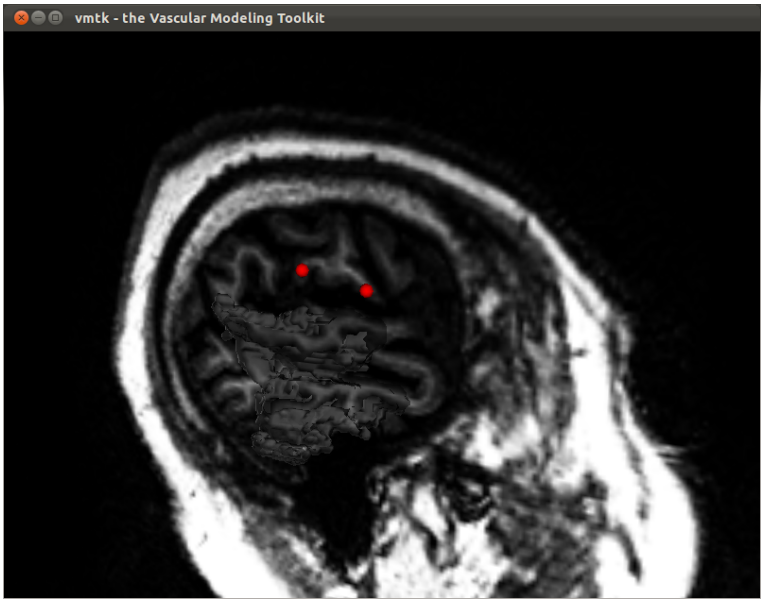


Figure 3.3: Segmenting a brain by Colliding Fronts. The volume chosen by the level sets method can be seen as a light grey shade in the middle of the brain. This image is from our segmentation process.

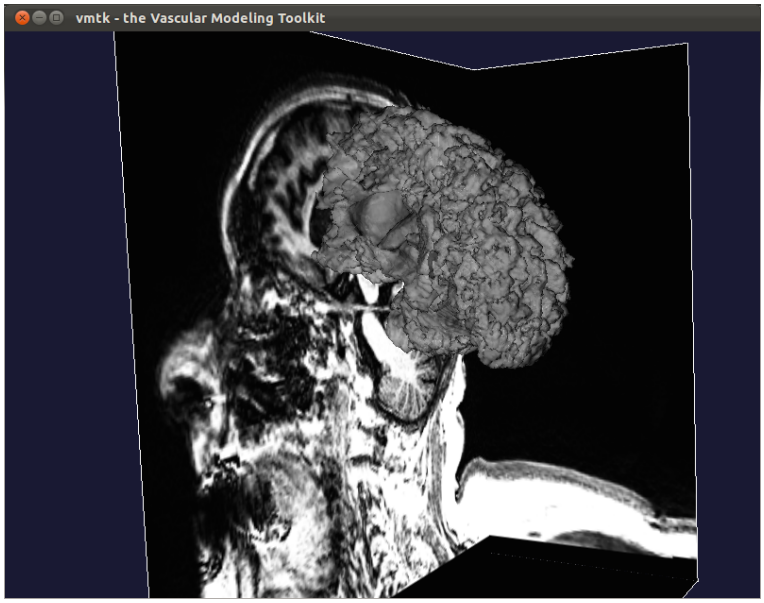


Figure 3.4: Level sets for half a brain.

some of the areas have different intensity compared to the same brain tissue in other areas. Particularly the ventricles have areas of CSF that have the same pixel values as the brain parenchyma around the sulci. Around the sulci, the interval for pixel values is $[9, \infty]$ Around the ventricles the interval is $[25, \infty]$ The intervals used are slightly wider than what is considered to be the parenchyma values. This is done because the algorithm did not always include the outer edges of the brain parenchyma in the level sets.

Smoothing

Based on the level set for the entire brain, the command `vmtkmarchingcubes` [25] is used to create a polyhedral surface mesh of the brain parenchyma. Such a surface can be seen in Figure 3.5. We need to transform this surface mesh into a volume mesh. To transform it, the surface has to be smoothed out and simplified. This simplification is necessary because VMTK simply can not handle too complex structures when creating a volume mesh. It is primarily designed for blood vessels, and the brain is generally more complex. To begin with, the surface is smoothed using the `vmtksurfacesmoothing` command in VMTK with a passband value of 0.1. A passband is the interval of frequencies which can pass through a filter without any smoothing being done. In our case this passband smooths out the rougher edges of the brain structure. The passband removes the largest irregularities, without creating big changes in the structure. The smoothed surface can be seen in Figure 3.5.

Simplification

By simplifying the mesh we also reduce the computation time required to run simulations. The mesh consists of faces and vertices in large numbers, up to a factor of 10^6. As the number of faces and vertices grow, so does the computational demands. Smoothing and simplifying the mesh will reduce the number of faces and vertices. The goal during the simplification process is to keep the major characteristics of the surface intact, while decreasing the complexity of the mesh.

Meshlab is system for the processing and editing of unstructured 3D triangular meshes [1]. The following filters in Meshlab are used to simplify the surface mesh:

- `Remove isolated pieces` ensures that there are no isolated pieces outside of the brain surface, or in the ventricles.
- `Merge Close Vertices` is used to decrease the resolution and the number of vertices and edges.
- `Surface Reconstruction` recreates the surface after the previous filter, as the previous surface might not be manifold, i.e., the surface might be damaged during the filtration.

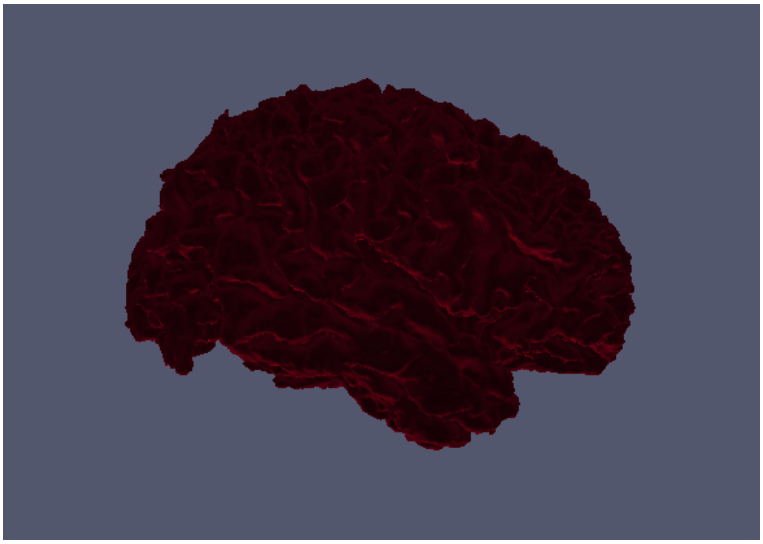
First the isolated pieces of brain tissue are removed. After that, merging of close vertices decreases the resolution. That does on some occasions cause holes to appear in the surface, thereby making the surface non-manifold. By reconstructing the surface, these problems were eliminated. Repeated use of the last two filters causes the size of the mesh to decrease gradually. In the end, the number of vertices and faces on the surface is reduced by a factor of 10, possibly more.

Mesh generation

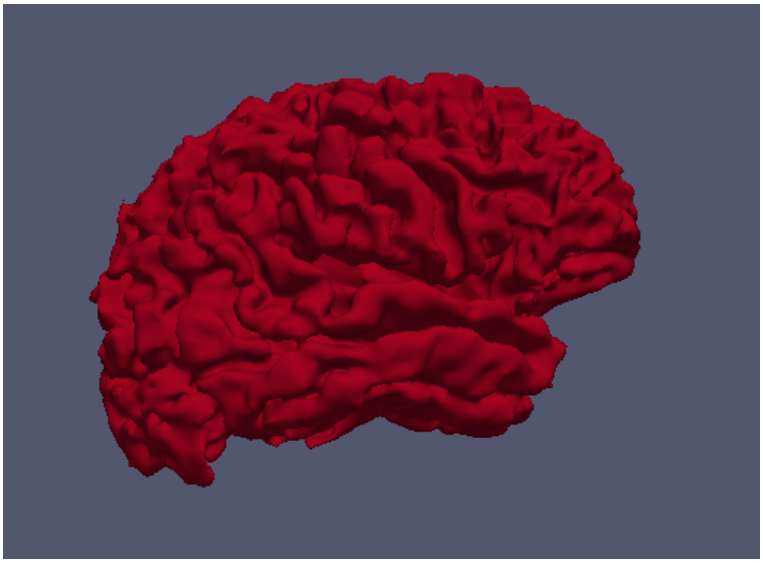
To create the final volume mesh, the `vmtkmeshgenerator` in VMTK is used. Varying the `edgelength` parameter leads to different number of faces and vertices, with higher edge length corresponding to a lower number of faces and vertices. The edgelength is the absolute nominal length of a surface triangle edge. This results in a mesh consisting of faces shaped like triangles. To convert it to a format compatible with DOLFIN [24, 23], a C++/Python package we use in our simulations, the `vmtkmeshwriter` is used to create the three-dimensional volume mesh.

3.3 ITK-Snap

As VMTK is primarily designed for use on segmenting blood vessels we considered an alternative software for segmenting the brain. However, this software



(a) Surface of a segmented brain, as seen from the left side.



(b) Surface of a smoothed brain, as seen from the right side.

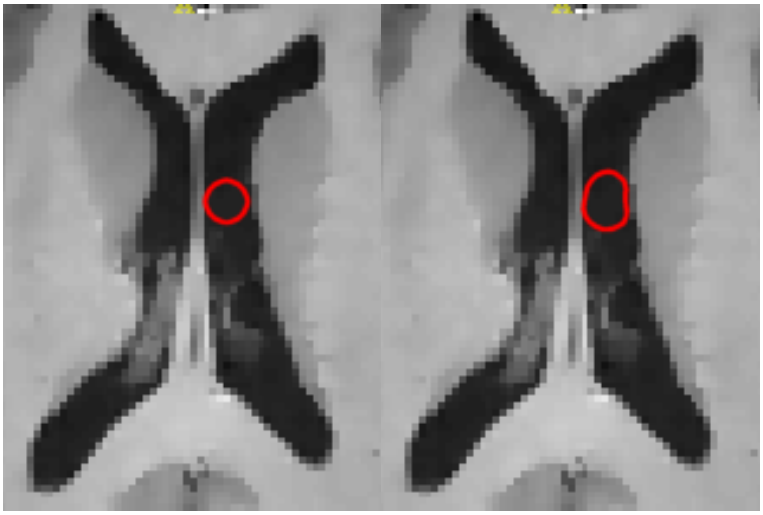
Figure 3.5: A segmented and smoothed brain.

was rejected due to problems described further down. ITK-Snap [39] is a software designed to segment 3D structures in medical images. Its most relevant feature for a quick and accurate segmentation of the brain is its Snake Evolution algorithm. The Snake Evolution algorithm consists of making a set of closed surfaces in 3D, and having them propogate automatically outwards, expanding the area. The area is defined within an interval of pixel values in the same manner as VMTK. This expansion is demonstrated in Figure 3.6.

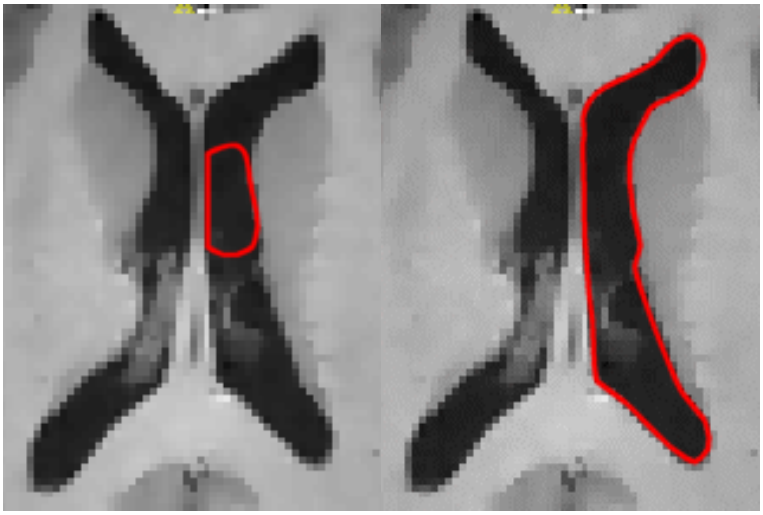
The speed of the expansion can be dependent upon up to three different kinds of velocities.

- The **propagation** velocity depends on how homogenous the image is. The more similar the pixel value, the faster. Unit speed is defined as the velocity at a completely homogenous image.
- A major problem with the Snake Evolution algorithm is that sometimes the snake leaks into parts of the structure that you do not wish to segment. The **curvature** velocity acts inwards on the snake, and is high at high curvatures of the structure. This helps to smooth out curves, and decreases the chances of leakage.
- The **advection** velocity is defined as the dot product of the unit vector perpendicular to the snake and the gradient vector of the feature image. In effect, this velocity can cause the snake to slow or even stop as it approaches edges on the image.

As mentioned, the leakage is a major problem. Especially on images with a lower contrast between the different tissues. Unfortunately our MRI images did not have a great contrast between tissues, so ITK-Snap is not used for the brain segmentation. However, for structures such as ventricles, ITK-Snap is very efficient and easy to use.



(a) Snake evolution, step one. (b) Snake evolution, step two.



(c) Snake evolution, step three. (d) Snake evolution, step four.

Figure 3.6: The snake evolution. Figures taken from *www.itksnap.org*

Chapter 4

Data Analysis

In the remainder of this thesis we will use the following abbreviations.

VS = Ventricular System, SAS = Subarachnoid Space, ICP = Intracranial Pressure, NPH = Normal-pressure Hydrocephalus, iNPH = idiopathic Normal-pressure Hydrocephalus, sNPH = secondary Normal-pressure hydrocephalus, AUC = Area Under the Curve

Background

Findings by, e.g., Matsumoto et al. [26] suggest that the pulsatile pressure in NPH patients differs from that in normal patients. In particular, the difference between systolic and diastolic pressure is significantly larger (X Pascal) as compared to normals (Y Pascal). However, work by, e.g., Eide et al. [11], considered the amplitude, risetime and risetime coefficient of the pressure measured in VS and SAS and no difference of transmantle gradients were found. These parameters will be more thoroughly defined further down. This chapter, in addition to attempting to replicate those results, extends previous studies by considering a more robust method, namely the differences in the area under the curve of the pressure pulse in the ventricular and subarachnoid space in 13 patients. 10 of the patients are the same as considered in Eide's [11] work, while the three others are patients with secondary NPH. Patient 1 had a break in the signal. The recording of patient 1 resumed after approximately five minutes, and the second signal is designated with *v2* in tables and results.

The AUC is an estimate for the total pressure during a cardiac cycle. Our goal in this chapter is to see if there is a gradient in the pulsatile pressure when the AUC is utilized as a measure. We will also attempt to extract a typical curve for each patient, based on average values for various criteria that will be explained in detail further down. AUC is given a subscript to determine whether it is for SAS or VS.

4.1 Analysis of the patient-specific data

Our patient-specific pressure data consists of recordings varying from 8, up to more than 24 hours long, differing amongst the patients. For each patient there is one measurement from the VS and one measurement from the SAS. The first two hours and the last hour of the recorded signals are excluded from the analysis. The exclusion is done due to possible interference and noise during the insertion and removal of the sensors. The pressure caused by a cardiac cycle is identified by the diastolic time (DT) and systolic time (ST) step indices. These represent the local minima and maxima of the signal, see Figure 4.1. The data was received from Rikshospitalet.

Because of calibration, no absolute measurements are available for the data. In this thesis we will only consider the relative values, i.e., the differences in pressure. The pressure is recorded in mmHG. During the analysis the unit of pressure is converted to Pascal (Pa). This is done because Pa is a more suitable and flexible unit. As both mmHG and Pa are units of pressure, the conversion formula is simply $Pa = 133.3 \cdot mmHG$. The extreme points are determined in the following fashion, where M_i and m_i are, respectively, the maximum and minimum points number i.

$$\begin{aligned} M_i &= \max\left(f^n_{Loc}(min_{i-1}), f^n_{Loc}(min_{i-1}+1), f^n_{Loc}(min_{i-1}+2), ..., f^n_{Loc}(min_{i-1}+p)\right) \\ m_i &= \min\left(f^n_{Loc}(max_i), f^n_{Loc}(max_i+1), f^n_{Loc}(max_i+2), ..., f^n_{Loc}(max_i+q)\right) \end{aligned} \tag{4.1}$$

f^n_{Loc} is the curve spanned out by the linear interpolation of an entire recorded pressure signal, with $Loc = location$ being either VS or SAS and n being the patient number.

The numbers $p = 70$ and $q = 200$ are chosen after considering plots of the data sets. They are chosen to ensure that the correct extreme points aree chosen. Had p and q been too large it is possible that more than one maximum or minimum would have been included. This could have caused one cardiac cycle to be determined as *two* cardiac cycles. These values are chosen after observing plots of the recorded signals. With variations in those numbers there are minor variations in the numbers of local maxima and minima points. The extreme points themselves are apparently more or less the same. The variations are likely partly due to variations in the frequency of the heartbeat.

We choose to define a cardiac cycle as identified if its corresponding DT and ST step indices in both VS and SAS happened within 0.05 seconds of each other, i.e., that they are reasonably synchronized. All cardiac cycles that could not be completely identified in the signal from both VS and SAS are excluded from the analysis. These criteria excluded approximately 20 percent of the data. Each cardiac cycle is defined by the linear interpolation of the ICP signal. A cardiac cycle is represented as a vector that can be plotted as a curve. These curves are normalized with the pressure at the start of the

diastole set to zero. The AUC for a cardiac cycle is defined as the integral of this curve. The integral of the curve is calculated by the trapezoidal rule [21]. The AUC is demonstrated in Figure 4.2. An alternative definition of the same AUC is the average amplitude of the single wave multiplied with the length of the single wave. Note that the AUC is a more robust way to analyse the pressure during a cardiac cycle than the amplitude considered in, e.g., [11], [26]. It is more robust because it considers the pressure through the whole cycle, whereas the amplitude is only a sample of the cycle. Note also that the time step for the normalization differs in the ventricular and subarachnoid space.

The differences between the AUC_{VS} and AUC_{SAS} are quantified individually for each single wave. The quantification is done with respect to AUC_{VS}, AUC_{SAS}, $AUC_{VS} - AUC_{SAS}$ and $\frac{AUC_{VS} - AUC_{SAS}}{AUC_{VS}}$.The mean of these values is computed, as well as the standard deviation (SD) for the relative difference. The difference of a single wave for VS and SAS is displayed in Figure 4.2. The green area minus the red area is the difference of the AUC values of that particular single wave.

Two ways of considering the AUC were calculated. The first uses the diastolic time steps for the respective VS and SAS curves. These results are displayed in Table 4.1. The second version of AUC calculates both AUC values dependent upon the time steps for VS. This results in the curve for SAS being shifted downward, causing AUC_{SAS} to decrease. Both curves are then normalized to zero at the VS start index. This shift in the SAS curve is displayed in Figure 4.1, while the results can be seen in Table 4.2. One consequence of the normalization is that the whole SAS curve is shifted downwards. The difference of the second AUC definition is displayed in Figure 4.2, with the green area minus the red area being the AUC value of that particular curve.

The average values of the first definition of the AUC, the differences as $AUC_{VS} - AUC_{SAS}$ in numerical value and relative difference, the standard deviation and the number of cardiac cycles evaluated for each patient is presented in Table 4.1. The median values are also calculated, and there are no substantial differences. Statistical significance is estimated using Welch's t-test [36]. Welch's t-test is an adaption of Student's t-test [9]. It is designed to be used on two sample groups with possibly unequal variance and sample sizes to determine if there is a statistically significant different between the two sample groups. The results can be viewed in Table 4.5.

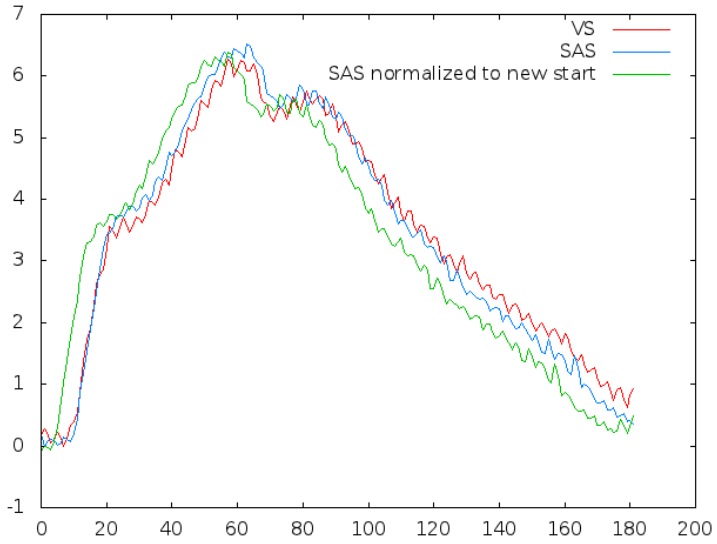
For comparison with Eide's previous work [11], [12], the average values of amplitude (dP), risetime (dT), risetime-coefficient (dR) and a variation on the risetime coefficient (dP/dT) are also calculated. Definitions can be found in Equation 4.2. Let $i = 1, 2, \ldots, n$ where n is the number of single waves. Our results correspond closely to his results.

Patient	AUC VS	AUC SAS	Difference	Difference in percent	SD	Single waves
(#)	(Pa s)	(Pa s)	(Pa s)	(%)	(%)	(#)
1	176,36	178,16	-1,80	-0,22	10,71	15 982
1 (v2)	221,14	225,93	-4,79	-1,68	6,14	53 030
2	553,88	558,95	-5,07	-0,86	4,78	59 427
3	381,61	387,76	-6,15	-0,79	13,73	75 433
4	309,29	310,34	-1,04	-0,17	4,80	77 921
5	596,63	585,08	11,56	1,46	53,82	65 675
6	205,57	209,02	-3,45	-2,54	122,79	55 193
7	351,24	352,08	-0,84	-0,26	6,27	50 417
8	341,61	337,92	3,70	1,15	7,85	62 643
9	318,40	317,40	1,00	0,50	5,72	83 784
10	405,81	394,47	11,33	2,56	10,05	26 378
11	258,73	263,19	-4,46	-1,63	5,75	87 805
12	139,80	140,43	-0,63	-0,01	11,73	83 446
13	163,17	166,40	-3,23	-2,11	5,35	166 031

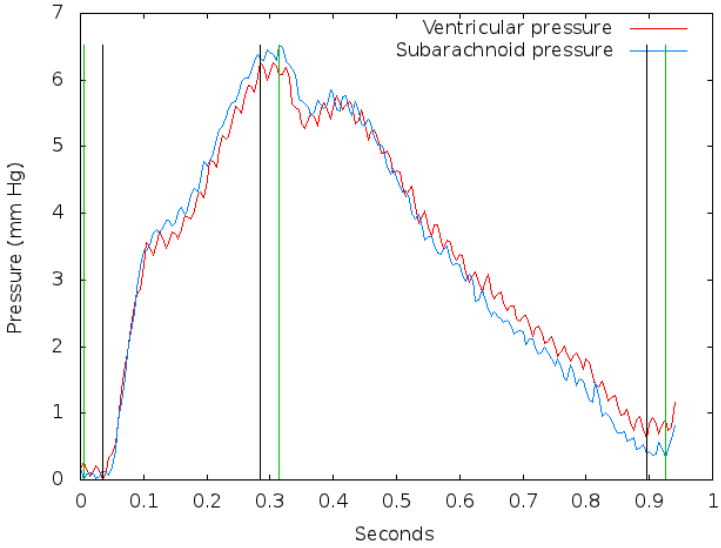
Table 4.1: AUC values, individual start and stop indices.

Patient	AUC VS	AUC SAS	Difference	Difference in percent	SD	Length
(#)	(Pa s)	(Pa s)	(Pa s)	(%)	(%)	(#)
1	175,96	173,29	2,67	2,75	11,6	15 982
1 (v2)	221,28	223,94	-2,67	-0,61	6,48	53 030
2	554,53	553,20	0,43	0,22	5	59 427
3	381,24	369,24	12,00	3,66	5,46	75 433
4	309,26	290,59	18,66	7,45	6,68	77 921
5	597,18	567,86	29,33	5,27	3,67	65 675
6	205,28	202,62	2,67	1,88	8,84	55 193
7	350,58	347,91	4,00	1,27	6,69	50 417
8	341,25	333,25	8,00	2,69	8,25	62 643
9	318,59	314,59	4,00	1,63	5,93	83 784
10	405,23	390,57	16,00	3,76	8,92	26 378
11	258,60	254,60	4,00	1,82	7,03	87 805
12	139,97	130,63	9,33	7,90	8,68	83 446
13	162,63	163,96	-1,33	-0,86	5,49	166 031

Table 4.2: AUC values, with both curves normalized at the VS time steps.



(a) The curves of a cardiac cycle in VS and SAS.



(b) The curves of a cardiac cycle in VS, SAS and SAS shifted downward.

Figure 4.1: Curves representing cardiac cycles.

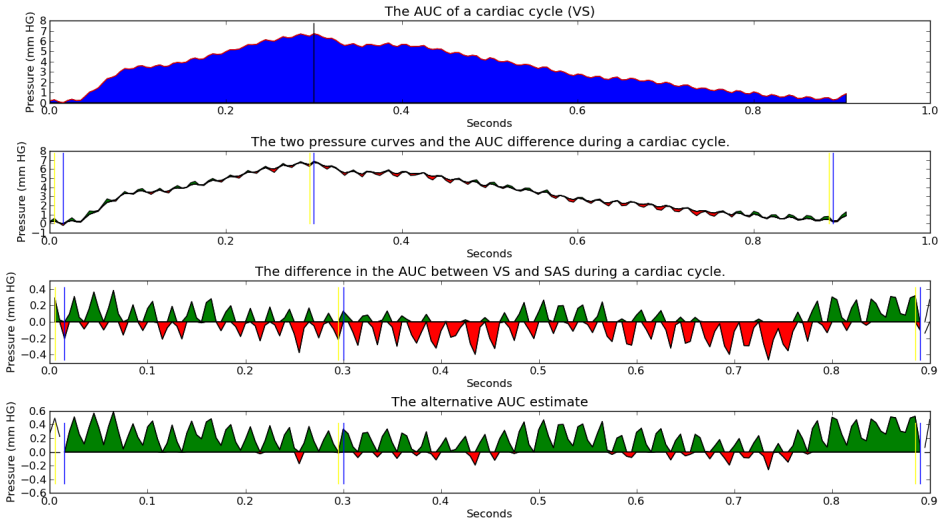


Figure 4.2: Graphics descriptions of AUC. The first plot is of the AUC of a cardiac cycle. The second is the pressure curves for VS and SAS for the same cardiac cycle with the AUC difference plotted in between. The third is of the AUC difference. The fourth is the difference for the second definition of the AUC.

Patient	dP	dT	dR	dP/dT
(#)	(Pa)	(s)	(Pa/s)	(Pa/s)
1	436,92	0,18	2 477,89	2 385,84
1 (v2)	529,88	0,20	2 597,94	2 587,34
2	1 245,60	0,28	4 564,97	4 492,41
3	844,44	0,28	3 011,64	2 991,44
4	606,43	0,28	2 161,99	2 158,79
5	1 194,92	0,26	4 632,89	4 589,70
6	562,18	0,20	3 379,74	2 878,86
7	722,26	0,28	2 707,88	2 614,71
8	716,97	0,28	2 663,61	2 598,42
9	794,33	0,26	3 061,30	3 028,34
10	811,52	0,25	3 337,72	3 231,84
11	618,61	0,25	2 534,05	2 479,33
12	396,41	0,13	3 442,02	2 949,81
13	451,26	0,18	2 744,05	2 455,66

Table 4.3: Values for VS.

$$
\begin{aligned}
dP_i &= f(max_i) - f(min_{i-1}) \\
dT_i &= \frac{max_i - min_{i-1}}{200} \\
dR_i &= \frac{dP_i}{dT_i} \\
dP &= \frac{\sum_{i=1}^{n} dP_i}{n} \\
dT &= \frac{\sum_{i=1}^{n} dT_i}{n} \\
dR &= \frac{\sum_{i=1}^{n} dR_i}{n} \\
\frac{dP}{dT} &= \frac{f(max_i) - f(min_{i-1})}{\frac{max_i - min_{i-1}}{200}}
\end{aligned}
\tag{4.2}
$$

The average of these values for all patients are presented in Table 4.3 and Table 4.4.

4.2 Filtering

As the signal recordings contain noise, we deemed it useful to filter the signal. A lowpass filter was provided by Svein Linge, and the pressure signals were run through it. The results are displayed in Table 4.6. Note that they are

Patient (#)	dP (Pa)	dT (s)	dR (Pa/s)	dP/dT (Pa/s)
1	439,44	0,18	2 513,98	2 407,08
1 (v2)	538,64	0,21	2 585,03	2 606,74
2	1 248,93	0,28	4 569,97	4 496,79
3	855,27	0,28	3 025,97	3 030,80
4	606,40	0,28	2 175,50	2 156,43
5	1 174,93	0,26	4 666,96	4 513,74
6	563,31	0,20	3 435,31	2 880,61
7	723,14	0,28	2 739,82	2 617,55
8	706,28	0,28	2 676,29	2 558,67
9	796,23	0,26	3 082,93	3 044,42
10	788,94	0,25	3 399,93	3 164,94
11	629,82	0,25	2 545,33	2 520,80
12	399,21	0,13	3 459,00	2 959,79
13	463,69	0,18	2 896,63	2 592,99

Table 4.4: Values for SAS.

Group	iNPH	sNPH
Mean	350.94	187.06
SD	133.22	62.98
SEM	40.16	36.36
N	11	3

Table 4.5: Results from Welch's t-test on iNPH vs sNPH patients, showing a statistically significant difference between the AUC values of sNPH and iNPH patients.

very similar to the results for the unfiltered signal. The average transmantle gradient points in the same direction for each patient in both the filtered and the unfiltered results. All the AUC values are slightly lower for the filtered signal, possibly due to noise removal.

There are also fewer cardiac cycles used in the filtered results. This could be because after the filtration, some of the cardiac could no longer be properly identified.

4.3 Results

Patients 1 through 10 have been diagnosed with idiopathic normal-pressure hydrocephalus (iNPH) while patients 11-13 have sNPH. There is a statistically significant difference between the sizes of the VS AUC between the sNPH

Patient	AUC VS	AUC SAS	Difference	Difference in percent	SD	Length
(#)	(Pa s)	(Pa s)	(Pa s)	(%)	(%)	(#)
1	135,07	137,10	-2,02	-1,00	10,32	8 536
1 (v2)	220,48	228,34	-7,86	-3,13	4,40	45 336
2	545,87	554,63	-8,75	-1,59	4,38	55 118
3	378,39	387,23	-8,85	-1,15	23,33	69 566
4	299,20	306,96	-7,76	-2,58	4,35	72 885
5	603,34	597,90	5,44	0,84	2,40	60 794
6	208,31	210,11	-1,80	-2,71	129,89	51 015
7	360,61	360,87	-0,26	-0,02	5,48	44 744
8	326,23	322,12	4,11	1,39	7,89	57 512
9	315,31	314,16	1,15	0,58	11,43	77 209
10	379,46	370,85	8,62	2,15	9,66	20 188
11	253,26	255,52	-2,26	-0,72	4,50	81 213
12	137,89	139,76	-1,88	-1,20	5,17	81 891
13	156,78	160,66	-3,88	-2,63	5,08	161 283

Table 4.6: AUC values after the signal recordings have been filtered. Note that they are very similar to the values from before the signal was filtered. There are no changes of direction in either for the transmantle gradients.

patients and the iNPH patients. The results from the t-test can be seen in Table 4.5, which displays the mean, standard deviation (SD), standard error of the mean (SEM) and the number of input parameters (N).

There is no statistically significant difference between the mean difference in percentage points for iNPH and sNPH patients. Patient 1 had a break in the signal that lasted for approximately five minutes. After the reboot, a new signal recording was initiated. Then there were notable changes in the pressure gradient for the average AUC. In total, 963 165 fully identified cardiac cycles were utilized in the analysis.

Six of the ten patients with iNPH had $AUC_{SAS} > AUC_{VS}$, while four had $AUC_{SAS} < AUC_{VS}$. All of the patients with sNPH had the highest AUC in SAS. However, these pressure gradients are all small, but several of the patients have a large standard deviation for the difference. The large SD indicates that there are some cycles with a notably larger pressure gradient. This indicates the presence of a sporadic pressure gradient in ICP. Whether these gradients are significant enough to cause changes in the brain parenchyma or not will be investigated in Chapters 8 and 9. The second version of AUC estimated has values that is lower or equal for AUC_{SAS}, while the values for AUC_{VS} remains the same. Equality would only happen for completely synchronized data signals. Compared to the first AUC values, the difference estimated in

percent rose by 1.1 (second signal patient 1, and patient 2) to 7.9 (patient 12) percentage points. For all patients except patient 13 and the second signal of patient 1 there is now a positive pressure gradient, i.e., higher pressure in VS than in SAS.

4.4 Summary

We have now investigated the pressure recordings of the patients. Our results coincide with previous work that show an increase in pulsatile ICP for patients diagnosed with iNPH, see Matsumoto [26] and Sklar et al. [32], but no substantial differences in the pulsatile pressure between VS and SAS, as in Eide et al. [11].

If one considers our second definition of the AUC, there is consistently such that $AUC_{SAS} < AUC_{VS}$. However, this is likely due to the mathematics of the definition of the second definition of the AUC. As the curve for the SAS pressure is normalized according to the value of the starting points for the VS curve, is is either normalized as low or *lower* than its lowest value. It is then unavoidable that its AUC value is weakly lower than it would be for the first definition of the AUC.

Chapter 5

The Finite Element Method

In this chapter we introduce the Finite Element Method (FEM). This is the main modeling tool we will use for our simulations. First the method will be presented. Then the method itself will be presented in four stages, as depicted in Figure 5.1. Some background information is also provided.

5.1 The Finite Element Method

The Finite Element Method is a method for creating numerical algorithms for approximating the solutions to Partial Differential Equations (PDE). A PDE is a system of differential equations involving the partial derivatives of a function, and are used to solve many problems that occur in the natural world. By enabling the user to choose between different spaces to solve the problem in, the FEM provides a framework for a huge variety of problems. The flexibility of the method makes it a very versatile tool. However, the generality of the method also makes it essential with human involvement to create the necessary algorithms. There are projects being pursued to generalize the method to decrease the need for human involvement, e.g., the FEniCS Project [22].

We will attempt to use the notation used by Brenner & Scott [7].

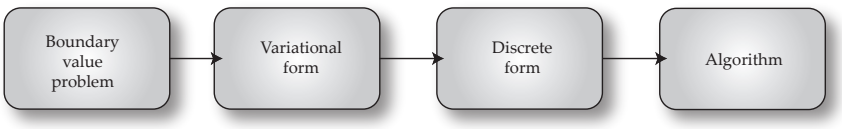


Figure 5.1: The four stages of the FEM.

5.2 Function spaces

Function spaces are topological spaces consisting of some set of functions under given constrains. These functions are defined from some space X to some space Y. One example of a function space is $L^1(\mathbb{R}) = \{u : \int_{\mathbb{R}} |u| dx < \infty\}$.

Sobolev Spaces

Function spaces are needed to formulate the PDE problems. These function spaces must be such that it is possible to find analytical estimates for errors, and that our solutions actually belong to these spaces. A reasonable degree of smoothness is also desired. Sobolev spaces have these qualities, and are used to solve PDEs. First we require the definition of the weak derivative [14].

Definition Suppose $u,\ v\ \in\ L^1_{loc}(\Omega)$ where Ω is an open and bounded subspace of $\mathbb{R}^n$ and α is a multiindex. We say that v is the α^{th}-weak partial derivative of u, written

$$D^\alpha u = v \tag{5.1}$$

provided

$$\int_\Omega u D^\alpha \gamma \, dx = (-1)^{|\alpha|} \int_\Omega v\gamma \, dx \tag{5.2}$$

For all test functions $\gamma \in C_c^\infty(\Omega)$.

The definition of a Sobolev space follows.

Definition The Sobolev space

$$W^{k,p}(\Omega) \tag{5.3}$$

consists of all locally summable functions $u : \Omega \mapsto \mathbb{R}$ such that for each multiindex α with $|\alpha| \leq k$, $D^\alpha u$ exists in the weak sense and belongs to $L^p(U)$.

One of the most frequently used Sobolev spaces is $W^{1,2}(\Omega) = H^1(\Omega) = \{v : \Omega \mapsto \mathbb{R}\ :\ \int f(x)^2 + (f'(x))^2 \, dx \leq \infty\}$. We will use $H^1(\Omega)$ as our trial space, with the additional requirement that $u \in H^1(\Omega)$ fulfills the Dirichlet boundary condition of the original problem. This space is usually called $H_0^1(\Omega)$. The letter H is used to denote these spaces because they are also Hilbert spaces [14].

Discrete Spaces and Finite Elements

Continuous spaces cannot be implemented in numerical methods. They must therefore be discretized. Consider also that we have some set Ω, which in the case of our simulation of the brain requires a 3D mesh of the brain. We choose to run our simulations using the FEM, for the reasons given above, and so we will need *finite elements*. The following definition is Ciarlet's definition of a finite element [8].

Definition Let

- $K \in \mathbb{R}^n$ be a bounded closed set with nonempty interior and piecewise smooth boundary (the *element domain*)
- P be a finite-dimensional space of functions on K (the space of *shape functions*) and
- $\aleph = \{N_1, N_2, ..., N_k\}$ be a basis for P (the set of *nodal variables*).

Then $(K, P, \aleph)$ is called a finite element. It is assumed that $\aleph$ lie in the dual space of some larger function space.

K is usually a two-dimensional triangle or three-dimensional tetrahedral, and that is what it will be in our simulations. We will use first degree Lagrange elements. Then we will have P as the the set of linear polynomials. We could use polynomials with higher degrees, but the computational power required for that is not worth the higher convergence rate of the solution.

5.3 Formulation

The FEM can be divided into four steps. The first is setting up the boundary value problem. The boundary value problem is the system of partial differential equations and the boundary conditions. The second step is to transform it to its equivalent variational form. In the third step the variational form is discretized. Finally, in the fourth step an algorithm for solving the problem is devised. In this section, all four of these steps will be described in detail, with the Poisson equation used as an example.

Boundary Value Problem

The boundary value problem is the system of PDEs, and corresponding boundary values. It is necessary that the problem is well-posed, i.e., that there exists a unique solution that is stable. A solution to the boundary value problem is therefore a solution to the PDE that also satisfies certain conditions regarding

the boundary. The boundary value problem for the Poisson equation is given by

$$\begin{cases} -\Delta u(x) &= f(x) \text{ in } \Omega \\ u &= u_0 \quad \text{on} \quad \partial\Omega_{\mathrm{D}} \\ -\nabla u \cdot n &= g(x) \quad \text{on} \quad \partial\Omega_{\mathrm{N}} \end{cases} \tag{5.4}$$

Where u and f are a functions that take $x \in \Omega \subset \mathbb{R}^n$ as a variable and returns a value in $\mathbb{R}$. On the two parts of the surface, $\partial\Omega_{\mathrm{N}}$ and $\partial\Omega_{\mathrm{D}}$ we have, respectively, the boundary values $g : \partial\Omega_{\mathrm{N}} \mapsto \mathbb{R}$ and $u_0 \in \mathbb{R}^{n-2}$. Here $\nabla u \cdot n$ is the directional derivative of u on the surface, pointing outwards. The two parts of the surface are non-overlapping and their subset union is all of Ω, i.e.,

$$\begin{cases} \partial\Omega_{\mathrm{N}} \cup \partial\Omega_{\mathrm{D}} &= \Omega \\ \partial\Omega_{\mathrm{N}} \cap \partial\Omega_{\mathrm{D}} &= \varnothing \end{cases} \tag{5.5}$$

The boundary on $\partial\Omega_{\mathrm{D}}$ is the Dirichlet boundary condition, also known as a first-order boundary condition. The boundary on $\partial\Omega_{\mathrm{N}}$ is the Neumann boundary condition, also known as the second-order boundary condition. Second-order because it corresponds to the first derivative.

Variational Form

The variational formulation, or weak form, of a problem is a reformulation of the original boundary value problem to a form more easily discretized and implemented on a computer. It is also done to avoid second-order derivatives. This step is usually done by hand, and does not require much computation. Once again the Poisson equation will be used to demonstrate the principles.

First we define a space $\hat{V} = \{v \in L^2 : v = 0 \text{ on } \partial\Omega_{\mathrm{D}}\}$. Here $\hat{V}$ is the *test space*. The functions v in $\hat{V}$ are called test functions. We now multiply our PDE with a test function.

$$-\Delta u(x) \cdot v(x) = f(x) \cdot v(x) \tag{5.6}$$

Then we integrate on both sides over Ω.

$$\int_{\Omega} -\Delta u(x) \cdot v(x)\, dx = \int_{\Omega} f(x) \cdot v(x)\, dx \tag{5.7}$$

Green's Theoreum [14] is used on the equation to remove any double derivatives.

$$\begin{aligned}\int_{\Omega} -\Delta u(x) \cdot v(x)\, dx &= \int_{\Omega} \nabla u(x) \cdot \nabla v(x)\, dx - \int_{\partial\Omega} (\nabla u(x) \cdot n) v(x)\, dS \\ &= \int_{\Omega} \nabla u(x) \cdot \nabla v(x)\, dx - \int_{\partial\Omega_{\mathrm{D}}} (\nabla u(x) \cdot n) v(x)\, dS \\ &- \int_{\partial\Omega_{\mathrm{N}}} (\nabla u(x) \cdot n) v(x)\, dS \\ &= \int_{\Omega} f(x) v(x)\, dx\end{aligned} \tag{5.8}$$

As v disappears where there is a defined Dirichlet boundary condition, the integration term of $\partial\Omega_D$ disappears, and we are left with the following equation after inserting $-\nabla u \cdot n = g(x)$ from Equation 5.4:

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x)\, dx + \int_{\partial\Omega_N} g(x) v(x)\, dS = \int_{\Omega} f(x) v(x)\, dx. \tag{5.9}$$

We then move all the parts of the equation that have both u and v as factors to the right side, and the parts with only v to the left side. In the above case, we are left with the following:

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x)\, dx = \int_{\Omega} f(x) v(x)\, dx - \int_{\partial\Omega_N} g(x) v(x)\, dS. \tag{5.10}$$

Which leads us to the variational formulation of the boundary value problem: For some function space V, find $u \in V$ such that

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x)\, dx = \int_{\Omega} f(x) v(x)\, dx - \int_{\partial\Omega_N} g(x) v(x)\, dS \;\; \forall\, v \;\in\; \hat{V}. \tag{5.11}$$

Discrete formulation

In the variational formulation, V and $\hat{V}$ are infinitely-dimensional function spaces. In order to implement the problem on a computer we need to discretize the spaces. Discretizing is done by limiting the dimension of the spaces to some finite number, creating new subspaces. Our domain Ω is divided into elements. These elements are usually triangular, and are indexed by some parameter $h > 0$. The new spaces, V_h and $\hat{V}_h$, will consist of piecewise polynomials of a

finite degree on these elements.

$$\begin{aligned} u_h &\in V_h \subset V \\ v &\in \hat{V}_h \subset \hat{V} \end{aligned} \tag{5.12}$$

Which gives us the discrete variational formulation of the boundary value problem: For some function space V_h, find $u_h \in V_h \subset V$ such that

$$\int_\Omega \nabla u_h(x) \cdot \nabla v(x)\, dx = \int_\Omega f(x)v(x)\, dx - \int_{\partial\Omega_N} g(x)v(x)\, dS \;\; \forall\, v \;\in\; \hat{V}_h \subset \hat{V}. \tag{5.13}$$

Algorithm

Implementing the discrete formulation on a computer is the next step. For that, an algorithm is required. As we now have discrete spaces, the spaces V_h and $\hat{V}_h$ each have a basis, of size m and n respectively.

$$\begin{aligned} V_h &= \text{span}\{\phi_i\}_{i=1}^n \\ \hat{V}_h &= \text{span}\{\hat{\phi}_j\}_{j=1}^m \end{aligned} \tag{5.14}$$

We can then define $u_h = \sum_{i=1}^{n} U_i\phi_i$. We then choose $v = \hat{\phi}_j$, and insert into the weak formulation.

$$\int_\Omega \nabla \sum_{i=1}^{n} U_i\phi_i \cdot \nabla\hat{\phi}_j\, dx = \int_\Omega f(x)\hat{\phi}_j\, dx - \int_{\partial\Omega_N} g(x)\hat{\phi}_j\, dS \text{ for } i = 1, 2, \ldots, n \tag{5.15}$$

As this relation holds for all $v \in \hat{V}_h$, summing over all the basis functions provides us with the following equation.

$$\int_\Omega \sum_{i=1}^{n}\sum_{j=1}^{m} \nabla U_i\phi_i \cdot \nabla\hat{\phi}_j\, dx = \int_\Omega \sum_{j=1}^{m} f(x)\hat{\phi}_j\, dx - \int_{\partial\Omega_N} \sum_{j=1}^{m} g(x)\hat{\phi}_j\, dS \tag{5.16}$$

Moving the summation signs and U_i outside of the integrals, leaves us with

$$\sum_{i=1}^{n}\sum_{j=1}^{m} U_i \int_\Omega \nabla\phi_i \cdot \nabla\hat{\phi}_j\, dx = \sum_{j=1}^{m} \int_\Omega f(x)\hat{\phi}_j\, dx - \sum_{j=1}^{m} \int_{\partial\Omega_N} g(x)\hat{\phi}_j\, dS. \tag{5.17}$$

Now this can be simplified to a more familiar matrix form.

$$\begin{aligned} A_{i,j} &= \int_{\Omega} \nabla\phi_i \cdot \nabla\hat{\phi}_j \, dx \\ b_j &= \int_{\Omega} f(x)\hat{\phi}_j \, dx - \int_{\partial\Omega_N} g(x)\hat{\phi}_j \, dS \\ U &= U_i \\ U &= A^{-1}b \end{aligned} \tag{5.18}$$

The matrix form is easily implemented on a computer, and thus an algorithm for solving a PDE has been designed. For the sake of implementation, it is customary for $\hat{V}_h$ and V_h to have the same dimension, making A a square matrix.

Abstract formulation

Abstract formulation of FEM problems can be advantageous for focusing on particular characteristics of the problems, such as error estimates, existence and uniqueness. There are three requirements for an abstract formulation.

- A Hilbert space V in which we search for a solution. Such a space also requires an inner product, $(\cdot,\cdot)_V$ and a norm, $||\cdot||_V$.
- A bilinear form $a(u,v):\ V \times V \mapsto \mathbb{R}$.
- A linear form $L(v):\ V \mapsto \mathbb{R}$.

Note that here $V = \hat{V}$. In the case of the Poisson equation, all of these criteria are fulfilled.

- The Hilbert space $H^1(\Omega)$.
- $a(u,v) = \int_{\Omega} \nabla u(x) \cdot \nabla v(x) \, dx$.
- $L(v) = \int_{\Omega} f(x)v(x) \, dx - \int_{\partial\Omega_N} g(x)v(x) \, dS$.

This leads to the abstract formulation of a FEM problem: Find $u \in V$ such that $\forall v \in V$

$$a(u,v) = L(v) \tag{5.19}$$

Generalizing the abstract form for the discrete case is straightforward: Find $u_h \in V_h$ such that $\forall v \in V_h$

$$a(u_h,v) = L(v) \tag{5.20}$$

Combining these two parts leads to the Galerkin orthogonality.

$$\begin{aligned} a(u,v) - a(u_h,v) &= L(v) - L(v) \ \ \forall v \in V_h \\ a(u-u_h,v) &= 0 \ \ \forall v \in V_h \end{aligned} \tag{5.21}$$

One can also express the matrix form in terms of the abstract formulation.

$$\begin{aligned} A_{i,j} &= a(\phi_i, \phi_j) \\ b_j &= L(\phi_j) \end{aligned} \tag{5.22}$$

Error estimates, uniqueness and existence

Using the Galerkin orthogonality, one can find several useful error estimates. The following theorem [13] presents a basic a priori error estimate

Theorem 5.3.1. *If u and u_h satisfy $A(u_h, v) = L(v)$ and $(a(u - u_h, v) = 0$, then $\forall v \in V_h$,*

$$||u - u_h||_V \leq \frac{\kappa_2}{\kappa_1} ||u - v||_V \tag{5.23}$$

Proof. Using ellipticity of V, continuity and bilinearity of a, along with Galerkin orthogonality gives us that $\forall v \in V_h$,

$$\begin{aligned} \kappa_1 ||u - u_h||_V^2 &\leq a(u - u_h, u - u_h) \\ &= a(u - u_h, u - u_h) + a(u - u_h, u_h - v) \\ &= a(u - u_h, u - v) \\ &\leq \kappa_2 ||u - u_h||_V ||u - v||_V \\ \Rightarrow ||u - u_h||_V \leq \frac{\kappa_2}{\kappa_1} ||u - v||_V \quad \square \end{aligned} \tag{5.24}$$

Stating uniqueness and existence of a problem is necessary. The most common way to state uniqueness and existence is through the Lax-Milgram theorem [14].

Theorem 5.3.2 (Lax-Milgram)**.** *Given the following conditions, with $\alpha, C, \beta > 0$,*

- *V is a Hilbert space with a norm and inner product.*
- *$|a(u,v)| \leq \alpha ||u||_V ||v||_V \ \ \forall v \in V$*
- *$|L(v)| \leq C||v||_V$*
- *$||u||^2 \leq \frac{1}{\beta} |a(u,u)|$*

There exists some unique $u \in V$ such that

$$a(u,v) = L(v) \ \ \forall v \in V \tag{5.25}$$

5.4 Summary

In this chapter we discuss the Finite Element Method. We go through the process of implementing the FEM, and include an important theorem for uniqueness and existence, namely the Lax-Milgram theorem. We provide the groundwork for working with linear elasticity.

Chapter 6

Linear Elasticity

Our final simulations will apply linear elasticity on a mesh of the human brain. The goal is to estimate how the brain matter behaves when certain pressure conditions are applied.

Elasticity is defined as a physical property of materials which return to their original shape after the stress that caused their deformation is no longer applied.

We will not consider time as a variable in our model. The reason for this is that we assume a quasi-static model, where the factor that depends on time is small enough that it can be neglected.

The mathematical model is from Brenner & Scott [7].

6.1 Linear Elasticity

Linear elasticity is a model of matter becoming deformed and strained when put under some predetermined pressure. The linearity springs from a linear relationship between stress and strain, defined through Hooke's Law. The elastic material is assumed to be isotropic and homogeneous. It is useful for simulations in, e.g., structural engineering.

A deformation of a medium is the change of position for a body from its initial configuration to its current configuration. Strain is a normalized measure of the deformation. Stress is a measure of the internal forces in play within the body.

Linear elasticity is most appropriate for small deformations. When the stress grows beyond the *yield strength* of a material, the stress-strain relationship is no longer linear. At this point plastic, i.e., permanent, deformations may occur. This point is defined as the von Mises yield criterion. It can be defined in terms of the von Mises stress. The von Mises yield criterion varies between different materials.

The von Mises stress, also known as equivalent stress, is defined as

$$\sigma_e = \sqrt{\frac{1}{2}[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]}, \tag{6.1}$$

where σ is the stress tensor and σ_i for $i = 1, 2, 3$ are the principal directions of the stress tensor.

We will now work through the steps of the FEM formulation for linear elasticity, see Figure 5.1.

6.2 Boundary Value Problem

In order to define the boundary value problem, we will first make some definitions. Let Ω be an open and bounded domain with boundary $\partial\Omega = \partial\Omega_N \cup \partial\Omega_D$. Let $u : \Omega \to \mathbb{R}$ be the displacement vector of the elastic material. Let σ be the stress tensor. Let f be the body forces of $\partial\Omega$ and g the pressure on the Neumann boundary. Let $\epsilon(u) = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$ be the strain tensor, and μ and λ the Lamé parameters.

The following set of equations defines the boundary value problem for linear elasticity, where the first equation define *Hooke's Law* in three dimensions [15].

$$\begin{cases} \sigma & = \lambda \mathrm{tr}(\epsilon(u))\, I + 2\mu\epsilon(u) \quad \text{in } \Omega \\ -\mathrm{div}(\sigma) & = f \qquad \text{in } \Omega \\ u & = 0 \qquad \text{on } \partial\Omega_{\mathrm{D}} \\ \sigma \cdot n & = g \qquad \text{on } \partial\Omega_{\mathrm{N}} \end{cases} \tag{6.2}$$

For some of our simulations we will be required to divide $\partial\Omega_D$ into several non-overlapping boundaries, but this does not require any extra calculations.

An elasticity boundary value problem is said to be *pure traction* if $\partial\Omega_D = \varnothing$ and *pure displacement* if $\partial\Omega_N = \varnothing$.

6.3 Variational formulation

This is the second step of the FEM formulation. We will now define the variational form of the boundary value problem from the previous section.

We begin with the following equation, and go through the steps described in the previous chapter.

$$-\mathrm{div}(\lambda\, \mathrm{tr}(\epsilon(u))\, I + 2\mu\epsilon(u)) = f \tag{6.3}$$

Multiply with a test function $v \in V = H^1(\Omega)$ and integrate over Ω on both sides.

$$\int_\Omega -\mathrm{div}(\lambda\, \mathrm{tr}(\epsilon(u))\, I + 2\mu\epsilon(u))v dx = \int_\Omega f v dx \tag{6.4}$$

Integrate by parts on the divergence on the left side.

$$\int_\Omega (\lambda \operatorname{tr}(\epsilon(u))\, I + 2\mu\epsilon(u)) \colon \nabla v dx - \int_{\partial\Omega_N} \sigma(u) n \cdot v dS = \int_\Omega f v dx \tag{6.5}$$

Note that : is the tensor dot product.

Which leaves us with the weak form of the boundary value problem: Find $u \in H^1(\Omega)$ such that

$$\int_\Omega (\lambda \operatorname{tr}(\epsilon(u))\, I + 2\mu\epsilon(u)) \colon \nabla v dx - \int_{\partial\Omega_N} \sigma(u) n \cdot v dS = \int_\Omega f v dx \quad \forall v \in V \tag{6.6}$$

Existence and uniqueness of solution

Now that we have the variational form we need to see if our problem has a solution, and whether the solution, if it exists, is unique.

For this section, we define the abstract formulation with the bilinear and linear form as

$$\begin{aligned} a(u,v) &= \int_\Omega (\lambda \operatorname{tr}(\epsilon(u))\, I + 2\mu\epsilon(u)) \colon \Delta v dx \\ &= \int_\Omega \{2\mu\epsilon(u) \colon \epsilon(u) + \lambda div(u) div(v)\} dx \quad \forall v \in V \\ L(v) &= \int_{\partial\Omega_N} \sigma(u) n \cdot v dS + \int_\Omega f v dx \quad \forall v \in V \end{aligned} \tag{6.7}$$

The Lax-Milgram theorem, Theorem 5.3.2, is used to prove uniqueness and existence of solutions. The first three conditions are in the case of linear elasticity trivial, but the fourth, coercivity, is not. To prove coercivity, *Korn's Inequality* is required. The proofs for the results in this chapter can be found in Brenner & Scott [7]. It is also assumed that $\partial\Omega$ is smooth, and that Ω is a polygon.

We will now present the results and definitions necessary to prove coercivity for our boundary value problem 6.2.

Definition $\hat{H}^k(\Omega)$ is defined by

$$\hat{H}^k(\Omega) \colon = \{v \in H^k(\Omega) \colon \int_\omega v dx = 0, \int_\Omega \operatorname{rot}\ v dx = 0\} \tag{6.8}$$

Where rot is defined as rot $v = -\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1}$

These spaces are closed subspaces of $H^1(\Omega)$ for $k \geq 1$.

We will require the following inequality.

Theorem 6.3.1 (Korn's Inequality)**.** *There exists a positive constant α such that*

$$||\epsilon(v)||_{L^2(\Omega)} + ||v||_{L^2(\Omega)} \geq \alpha ||v||_{\hat{H}^1(\Omega)} \quad \forall v \in \hat{H}^1(\Omega) \tag{6.9}$$

Following from Korn's Inequality comes the following corollary, which we will need to prove coercivity of the boundary value problem 6.2.

Corollary 6.3.2. *Let $V = \{v \in H^1(\Omega) : v|_{\partial\Omega_D} = 0\}$ where the measure of $\partial\Omega_D > 0$. There exists a positive constant C such that*

$$||\epsilon(v)||_{L^2(\Omega)} \geq C||v||_{H^1(\Omega)} \quad \forall v \in V \tag{6.10}$$

And finally, from Corollary 6.3.2, coercivity follows directly, which is stated in the following theorem. Note that in our particular case, the function w is zero.

Theorem 6.3.3. *Assume that $f \in H^{-1}(\Omega)$, $h = w|_{\partial\Omega_D}$ where $w \in H^1(\Omega)$, $(\sigma(u)n)|_{\partial\Omega_N} \in L^2(\partial\Omega_N)$ and that the measure of $\partial\Omega_D$ is greater than zero. Then the variational problem defined by Equation 6.6 has a unique solution.*

However, some of the later simulations might be pure traction problems. We will therefore present results regarding the solution of pure traction problems.

Theorem 6.3.4. *Assume $f \in L^2(\Omega)$ and $\sigma(u)n \in L^2(\partial\Omega)$. Then the variational problem follows.*

Find $u \in H^1(\Omega)$ such that

$$\begin{gathered} a(u,v) = \int_\Omega f \cdot v dx + \int_{\partial\Omega} \sigma(u)n \cdot v dS \quad \forall v \in RM, \\ RM := \{v : v = c + b(x_2, -x_1)^t, c \in \mathbb{R}^2, b \in \mathbb{R}\} \end{gathered} \tag{6.11}$$

is a solution if and only if the following compability condition holds:

$$\int_\Omega f \cdot v dx + \int_{\partial\Omega} (\sigma(u)n) \cdot v dS = 0 \quad \forall v \in RM \tag{6.12}$$

When 6.11 is solvable, there exists a unique solution in $\hat{H}^1(\Omega)$.

Locking is a problem for some numerical approximations whose mathematical formulation has a parameter dependency [5]. In linear elasticity this problem comes from the parameter Poisson's ratio ν. More about that parameter in Chapter 7. Locking tends to result in poor convergence rates for the displacement vector u.

Theorem 6.3.5. *For fixed μ and λ, let $u \in \hat{H}^2(\Omega)$ satisfy the pure traction problem. Let $u_h \in \hat{V}_h$ satisfy*

$$a(u_h, v) = \int_\Omega f \cdot v dx + \int_{\partial\Omega} \sigma(u) n \cdot v dS \quad \forall v \in \hat{V}_h. \tag{6.13}$$

Then there exists a positive constant $C_{(\mu,\lambda)}$ such that

$$||u - u_h||_{H^1(\Omega)} \leq C_{(\mu,\lambda)} h ||u||_{H^2(\Omega)}. \tag{6.14}$$

6.4 Discretization

This is the third step of the FEM formulation. We need to discretize the formulation, as a step in the preparations for implementing it on a computer.

We reduce the test and trial space V to the discrete space V_h. So our variational formulation is now: Find $u_h \in V_h$ such that

$$\int_\Omega (\lambda \operatorname{tr}(\epsilon(u_h))\, I + 2\mu\epsilon(u_h)) : \Delta v dx = \int_{\partial\Omega_N} g v dS + \int_\Omega f v dx \quad \forall v \in V_h. \tag{6.15}$$

6.5 Algorithm

To create the algorithm, we consider the basis for $V_h = \operatorname{span}\{\hat{\phi}_i\}_{i=1}^n$ and the ansatz $u_h = \sum_{i=1}^n U_i \phi_i$. Choosing $v = \phi_j$ for some j, and inserting the ansatz we get the following equation.

$$\int_\Omega (\lambda \operatorname{tr}(\epsilon(\sum_{i=1}^n U_i \phi_i))\, I + 2\mu\epsilon(\sum_{i=1}^n U_i \phi_i)) : \Delta \phi_j dx = \int_{\partial\Omega_N} g \phi_j dS + \int_\Omega f \phi_j dx \tag{6.16}$$

Summing over all j and moving the summation signs outside of the integration leaves us with

$$\sum_{j=1}^n \sum_{i=1}^n U_i \int_\Omega (\lambda \operatorname{tr}(\epsilon(\phi_i))\, I + 2\mu\epsilon(\phi_i)) : \Delta \phi_j dx = \sum_{j=1}^n \int_{\partial\Omega_N} g \phi_j dS + \sum_{j=1}^n \int_\Omega f \phi_j dx \tag{6.17}$$

Now it is straightforward to define the algorithm

$$\begin{aligned} A_{i,j} &= \int_{\Omega} (\lambda \operatorname{tr}(\epsilon(\phi_i))\, I + 2\mu\epsilon(\phi_i)) : \Delta\phi_j dx \\ b_j &= \sum_{j=1}^{n} \int_{\partial\Omega_N} g\phi_j dS + \sum_{j=1}^{n} \int_{\Omega} f\phi_j dx \\ U &= A^{-1}b \end{aligned} \tag{6.18}$$

6.6 Boundary Conditions

There are four alternative boundary conditions that were considered.

- `Dirichlet` boundary, also known as an essential boundary condition, is the function u, i.e., displacement, fixed on some boundary. It also serves to lock the boundary boundary in place, and ensures a solution with the numerical methods.
- `Neumann` boundary, also known as a natural boundary condition, is the derivative of u, the pressure, fixed at some part of the boundary..
- `Robin` boundary, also known as impedance boundary condition, is a weighted combination of both Dirichlet and Neumann boundary conditions. It has the advantage of both setting a pressure, and semi-locking the boundary values of the solution in place.
- `Mixed` boundary means that part of the boundary is Neumann and the remaining part of the boundary is Dirichlet.

In our simulations we will consider a Robin boundary condition and mixed boundary values. A Robin boundary allows us to modify the extent to which the brain mesh will be locked in place, and also creates a smooth transfer between the two estimated pressure points. The alternative is to lock the brain in place with Dirichlet conditions on certain points of the mesh, and then add one or more Neumann boundary conditions on the remaining mesh. The Robin boundary condition is a more natural choice, as the brain is partly locked in place inside of the skull by the strains connecting pia mater and the arachnoid, yet still has room to be displaced. However, it might not be possible to implement it in an acceptable fashion.

Neumann boundary

A pure traction, i.e., only Neumann boundaries, is perhaps the most natural solution for our problem. However, we will then run into problems with out numerical methods, see Theorem 6.3.4.

Robin Boundary

The way we will implement a Robin boundary condition is by inserting it into the elasticity equation in the same fashion as a Neumann boundary condition. Namely

$$\int_{\partial\Omega} \sigma \cdot n dS \; = \; \int_{\partial\Omega} \frac{1}{k(x)} u + p \cdot n dS, \tag{6.19}$$

where p is the pressure on the surface, and $k(x)$ is some function that takes the spatial coordinates x as input. The k function is used to weight the displacement and pressure conditions, and thereby determine which of the two has the greatest influence on the boundary.

6.7 Summary

This chapter has been about the mathematical model of linear elasticity. The main focus has been to present the FEM for linear elasticity and show that there is a unique solution, given certain conditions.

We have also considered different boundary conditions. The choice of boundary conditions will be discussed further in Chapter 7 and 9, as they are a significant part of the final simulations. The choice of boundary values are influenced by the results in linear elasticity presented in this chapter, in particular Theorem 6.3.4.

Note the variational formulation for linear elasticity. We use this equation in the implementation.

$$\int_{\Omega} (\lambda \operatorname{tr}(\epsilon(u_h))\, I + 2\mu\epsilon(u_h)) \colon \Delta v dx = \int_{\partial\Omega_N} g v dS + \int_{\Omega} f v dx \quad \forall v \in V_h. \tag{6.20}$$

Chapter 7

Implementation

Many systems can be described by PDEs. However, solving these PDEs analytically is frequently impossible. Numerical methods rely on iterative methods for approximating the analytical solution. We will now prepare to implement the variational problem for linear elasticity defined by Equation 6.6.

7.1 FEniCS

The FEniCS Project [22] is a collection of open source software components. It is designed to automatically and efficiently generate code to solve PDEs, set boundary values, handle variational formulations and finite elements, as well as edit meshes. FEniCS can be used in high-level Python or low-level C++ code. One of the biggest advantages of FEniCS is that it removes to need for the user to take care of the minor details of the Finite Element Method. The user can instead focus on the big picture. However, it is still necessary to define the problem, and adapt the code to your own needs.

FEniCS can be used in both Python and C++. DOLFIN [24, 23] is FEniCS' main interface, and is written in C++ for efficiency reasons. A high-level user interface is generated through Swig [2].

The numerical results are provided as a *displacement vector*. The displacement vector is defined as the shortest distance from the initial position to the current position. The *deformation* is the transformation from the initial position of a shape to the current position.

In this thesis we mainly use two FEniCS programs. Both use linear elasticity. The first, `sphere.py`, is a very simplified model of the brain, which is simply a sphere inside a bigger sphere. A mesh of the spherical form has been provided. The spherical mesh is displayed in Figure 7.1. The spherical mesh us used to gain an idea of how great displacements and what kind of deformations one can expect with certain parameters, and to test the code up against an analytical solution for the one-dimensional problem. The second program, `brain.py` is used in the simulation of the brain. For this brain simulation,

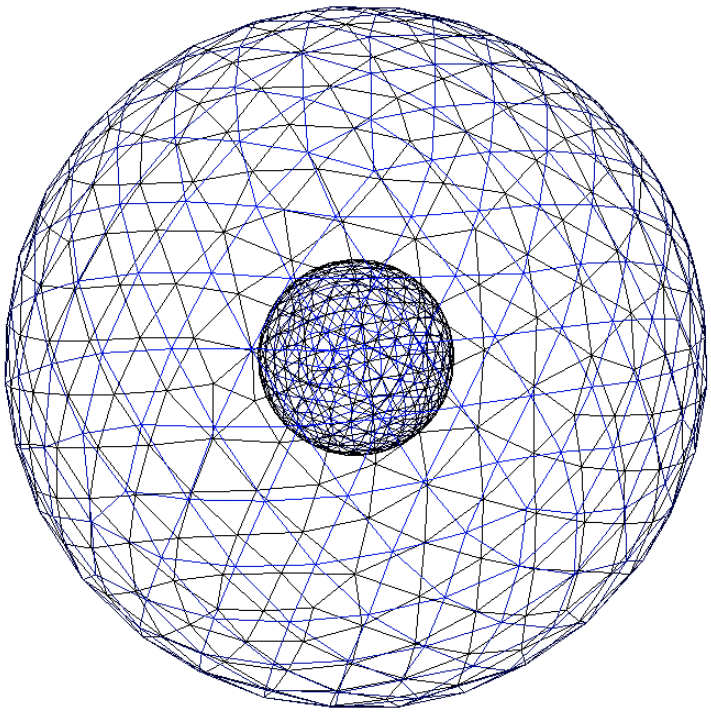


Figure 7.1: The mesh representing a sphere inside a sphere.

a generic brain mesh is used. Patient-specific pressure measures are used on this model. There are not many differences in the two codes, and these are pointed out.

The brain mesh can be seen in Figure 7.2. We would have prefered to use the most detailed brain mesh we created. However, the computational resources required are too great for it to be conveniently used. The more detailed mesh is also displayed in Figure 7.2.

Sphere

The script `sphere.py` is a very straightforward implementation of linear elasticity. It will either have a Robin boundary on the whole surface, or a Dirichlet condition on the outer surface and Neumann condition on the inner surface. The Robin boundary will be created such that the displacement part of the boundary condition is close to zero. This will serve to emulate a Neumann condition. There will be prescribed surface pressure on either the inner, outer or both spheres.

```
import sys
import numpy as np
from dolfin import *
```

Three Python packages are imported. The first, `sys`, is for reading command-line arguments. The second is for array implementation in Python. The third is DOLFIN, the main user interface to FEniCS.

The next step is importing the data to be used as pressure on the Neumann and Robin boundaries. There are two options. First is the difference between the two pressure signals, VS and SAS.

```
d1 = np.array([float(i) for i in \
open(sys.argv[3], 'r').readlines()])
d2 = np.array([float(i) for i in \
open(sys.argv[4], 'r').readlines()])
d1, d2 = d1-d1[0], d2-d2[0]
d3 = 133.3*(d1-d2)
```

In the first two lines above, the curves representing the pressure during the cardiac cycle are imported. In the third the two curves are normalized to zero pressure fixed at the start of the curve. The fourth line is the pressure difference between the pressure recorded in VS and the pressure recorded in SAS. The pressure is also converted from mmHG unit to a Pa unit by multiplying the pressure with the numerical conversion value 133.3.

The second alternative for pressure on the boundaries is to use either of the pressure curves.

```
d1 = np.array([float(i) for i in \
open(sys.argv[3], 'r').readlines()])
d2 = np.array([float(i) for i in \
open(sys.argv[4], 'r').readlines()])
d1, d2 = d1-d1[0], d2-d2[0]
def AUC(List):
    return sum(List) - (List[0] + List[-1])/2.0

auc = AUC(d2)/float(len(d2))*200
d2 = d2-auc
d1 = d1-auc
```

The first three lines are the same as above. After that, a function to estimate the AUC is defined. Then in the last three lines the pressure curve for SAS is normalized such that that its AUC, as calculated by the trapezoidal rule, is zero. Then the pressure curve for VS is normalized with the same numerical value.

For the rest of the program, we will only provide the code in such as way that it assumes we have one pressure curve. There are a few minor differences, but these are trivial.

```
mesh = Mesh("sphere_small.xml")
V = VectorFunctionSpace(mesh, "CG", 1)
Q = FunctionSpace(mesh, "CG", 1)

f = Constant((0,0,0))

g = Expression(("-t"), t = d3[0])
k = Expression("1000000.0")
n = FacetNormal(mesh)
```

In the first line we import the mesh described in Section 7.1.1. In the second and third lines we define a vector function space and function space, respectively, on the mesh. Linear Lagrange finite elements are created.

In the fifth line, the body forces are fixed at zero. In the three last lines we first define the pressure function g by the pressure curve. We then define the function k for the Robin condition, described in more detail in Chapter 6. Last the facet normal vector n is defined. It points directly outwards from the mesh surface.

```
boundary_markers = FacetFunction("uint", mesh)
boundary_markers.set_all(0)
```

All the surface elements are marked for later use.

```
class NeumannBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and \
        (x[0]**2 + x[1]**2 + x[2]**2) < 0.03**2 \
        + DOLFIN_EPS

neumann_boundary = NeumannBoundary()
neumann_boundary.mark(boundary_markers, 1)
```

The Neumann boundary is defined on the inner sphere in the class `NeumannBoundary`. In the last two lines the outer surface is marked as a Neumann boundary.

Creating a Robin boundary is essentially the same process. The only difference would be that in our case the class would return all values on the boundary.

```
def mesh_boundary(x, on_boundary):
    return on_boundary and \
    x[2]*x[2] + x[1]*x[1] + x[0]*x[0] > 0.065**2 \
    + DOLFIN_EPS

bc = DirichletBC(V, Constant((0,0,0)), mesh_boundary)
```

First we define a function that will return values on the outer sphere. In the next to last line we define the Dirichlet boundary.

```
ds = Measure("ds")[boundary_markers]
```

Here we define a surface integration dependent upon the boundary markers.

```
u = TrialFunction(V)
v = TestFunction(V)
u1 = Function(V)
```

Here we define the trial function u, the test function v and our solution function u1.

```
lmbda = float(sys.argv[1])
mu = float(sys.argv[2])

def sigma(v):
    return 2.0*mu*sym(grad(v)) + \
    lmbda*tr(sym(grad(v)))*Identity(v.cell().d)
```

In the first two lines we import the values of μ and λ from the command line. We then define the stress tensor σ.

```
a = inner(sigma(u), grad(v))*dx
L = inner(f, v)*dx + inner(g,dot(v, n))*ds(1)
```

We now define the variational form of our problem. Above is the variational formulation for mixed boundary conditions. Note the surface integration over the Neumann boundary in the last line. Below is the variational formulation for Robin boundary conditions.

```
a = inner(sigma(u), grad(v))*dx   \
- inner((1.0/k)*u, v)*ds(1)
L = inner(f, v)*dx + g*dot(v, n)*ds(1)
```

Here note the k function in the first line. As we defined it earlier to be 10^6, we see that the displacement part of the boundary condition will have a very small effect compared to the pressure condition.

```
file_name = sys.argv[5] + '_' + sys.argv[1] + \
'_' + sys.argv[2] + '.pvd'
u_file = File(file_name)
u_file << u1
```

In the first line we define the name of the .pvd files we use to save the results. In the second line a file is created, while in the third we save the displacement after zero time-steps.

```
t=0
while t < len(d1):
    g.t = d2[t]
    A = assemble(a)
    b = assemble(L)
    bc.apply(A,b)
    solve(A, u1.vector(), b)

    mesh.move(u1)
    t+=1
    u_file1 << u1
```

We set the time-step. We then proceed to loop through the entire cardiac cycle. Inside the loop we first update the pressure value in the variational

form. We then assemble the A matrix and b vector, see Chapter 5. Then we apply the Dirichlet boundary conditions, and proceed to solve the linear elasticity problem.

In the third to last line we update the mesh according to the displacements. In the second to last line we update the time step, and in the last line we save the displacement.

To run a single time-step of constant pressure, we need only add a simple `if` test with a `break` command if $t \geq 1$.

```
# strain tensor:
def epsilon(v):
    return 0.5*(grad(v) + grad(v).T)

# stress tensor:
def sigma(v):
    return 2.0*mu*epsilon(v) + \
    lmbda*tr(epsilon(v))*Identity(3)

def mean_pressure(sigma):
    return 1.0/3.0*tr(sigma)

def deviatoric_stress(sigma):
    return sigma - \
    mean_pressure(sigma)*Identity(3)

def von_Mises(v):
    sigma_ = sigma(v)
    dev_sigma = deviatoric_stress(sigma_)
    return sqrt(3.0/2*inner(dev_sigma, dev_sigma))

vonMises_proj = project(von_Mises(u1), Q)
stress_name = 'name.pvd'
stress_file = File(stress_name)
stress_file << vonMises_proj
```

Finally we prepare to estimate the von Mises stress. We do this by defining the strain tensor, stress tensor, mean pressure, deviatoric stress and finally von Mises stress. In the fourth to last line we project the von Mises stress, i.e., we create a format we can visualize. In the three last lines we define the name of a file to save the von Mises stress in, create a file, and finally save the stress.

Brain

`brain.py` is essentially the same program as `sphere.py`, except with minor differences in the boundary conditions. We will provide only the major differences, instead of going through much of the same code twice.

```
def center_boundary(x, on_boundary):
    tol = 1e-2
    return on_boundary and ((x[0]-0.105)**2 + \
    (x[1]-0.171)**2 + (x[2]-0.095)**2)**(1/2.0) \
    > 0.055 + DOLFIN_EPS
```

The center of the brain mesh is located at coordinates $(105, 171, 95)$. So we make small modifications on the Dirichlet boundary in order to get a circle around the center of radius 0.055 meters.

```
class NeumanBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1e-2
        return on_boundary and 0.055 + DOLFIN_EPS >= \
        ((x[0]-0.105)**2 + (x[1]-0.171)**2 \
        + (x[2]-0.095)**2)**(1/2.0)
```

We make a class in the same manner when creating the Robin boundary. Otherwise it is essentially the same code.

Slight modifications are made to the code to adapt to some of the different simulations.

These codes are partly based on the FEniCS demos provided on the FEniCS homepage [22]. A more detailed walk-through of FEniCS coding can be found there.

7.2 Parameters

There are a few different elasticity parameters in use in the literature. As these parameters can be defined by each other, when refering to parameters taken from an article, we will use the ones most convenient. These will be converted to the same parameters before the simulations.

- Young's modulus, E, is defined as the normal stress divided by linear strain [29] of a material. Young's modulus is a parameter for the stiffness of a material when put under linear strain, with a higher Young's

modulus meaning a stiffer material. Young's modulus is always positive. Young's modulus has the unit Pascal.

- Poisson's ratio, ν, is the measure of the negative transverse strain divided by the axial strain. It describes what happens when you apply pressure in one direction, and the material expands in other directions. This kind of expansion leads to ν having values in (0, 0.5] In the case of negative values for ν, the material is compressed in other directions. Negative Poisson's ratio has values in a ν value in $[-1, 0)$. Poisson's ratio has no unit.

- Lamé's first parameter, λ, is used because it simplifies a matrix in *Hooke's Law*. λ can take negative values, but is usually positive. It has the unit Pascal.

- Lamé's second parameter, or shear modulus, is usually denoted by either μ or G. It is defined by shear stress divided by shear strain. Also, μ is another parameter that describes the reaction of a material when put under strain. For μ, the strain modelled is shear strain, i.e. cutting. Its value is always positive, with the unit Pascal.

Young's modulus is usually paired up with Poisson's ratio, while the two Lamé parameters are usually paired up. Therefore, only the conversion formulae between the two sets of parameters follows. If you have two of the parameters, the others can be calculated.

$$\begin{aligned} \lambda &= \frac{E\nu}{(1+\nu)(1-2\nu)} \\ \mu &= \frac{E}{2(1+\nu)} \\ E &= \frac{\mu(3\lambda+2\mu)}{\lambda+\mu} \\ \nu &= \frac{\lambda}{2(\lambda+\mu)} \end{aligned} \tag{7.1}$$

In the literature there are different estimates for the elasticity parameters for the brain. These differences can be due to different methods for extracting the parameters, or differences in the brain tissue used in the processes.

- Dutta Roy et al. [10] used a non-linear model to investigate the size of the transmantle gradient required to create the ventricular growth associated with NPH. Their values are based on experiments. They used the following parameters in calculations.

  - An incompressible brain with Poisson's ratio equal 0.5 and Young's modulus set at 467.31 Pa.

Source	E (Pa)	ν	λ (Pa)	μ (Pa)
Dutta-Roy et al. [10]	467.31	0.5	$4.25e^{17}$	155.77
Dutta-Roy et al. [10]	464.19	0.49	7632.65	155.77
Dutta-Roy et al. [10]	420.58	0.35	363.46	155.77
Humphrey & Smith [33]	5000	0.4	7142.86	1785.71
Taylor & Miller [38]	584.4	0.35	505.04	216.44
Wirth & Sobey [35]	585.0	0.354	523.79	216.02
OCCAM-Fields-MITACS [6]	21600	0.35	18666.67	8000

Table 7.1: Elasticity parameters.

  - A nearly incompressible brain with Poisson's ratio equal 0.49 and Young's modulus set at 464.19 Pa.
  - A compressible brain with Poisson's ratio equal 0.35 and Young's modulus set at 420.58 Pa.

- Humphrey and Smith [33] uses a Poisson's ratio of 0.4 and Young's modulus of 5000 Pa. Their values are based on a review of the litterature.

- Tayler and Miller [35] refers a Poisson's ratio of 0.35 and Young's modulus of 584.4 Pa. Their values are based on experimentation.

- Wirth and Sobey [38] models the CSF distribution and brain displacement during an infusion test, and uses a Young's modulus of 585 Pa and a Poisson's ratio set at 0.354.

- A proceeding from OCCAM-Fields-MITACS Biomedical Problem Solving Workshop, 2009 [6] investigates transmantle gradients as a mechanism for hydrocephalus. They used a shear modulus, μ, of 8000 Pa and Poisson's ratio of 0.35.

Converting these parameter values according to the above formulae and we can see the results in Table 7.1.

7.3 Cardiac Cycles

We wish to use examples of cardiac cycles from our analysis in Chapter 4. Considering the values in Table 4.1, we have very similar results for all the patients. So we extract examples of curves that span the range of results. The curves we will use have the following characteristics, within a small error margin.

- Average AUC size and average AUC difference of the ten iNPH patients.
  - The average AUC_{VS} for the ten iNPH patients is 366.18 Pa s

  - The average AUC_{SAS} for the ten iNPH patients is 361.91 Pa s.
  - The average AUC difference in percents is 0.1 %. That is on average $AUC_{VS} = AUC_{SAS} \cdot 1.001$.

- Average AUC size and average AUC difference of the three sNPH patients.
  - Average AUC_{VS} for the three sNPH patients is 187.06 Pa s.
  - Average AUC_{SAS} for the three sNPH patients is 189.73 Pa s.
  - Average AUC difference in percents is -1.25 %. Which means that on average $AUC_{VS} = AUC_{SAS} \cdot 0.9875$.

- Large AUC size and $AUC_{VS} \gg AUC_{SAS}$.
  - A large AUC is chosen with AUC_{SAS} at approximately 150 % of the average AUC_{VS} size. 533.2 Pa s is chosen as this large value. Two of the iNPH patients had a greater AUC_{VS} average than this. None of the sNPH patients did.
  - The size difference between the AUC values is chosen at 5 %. This is twice the size of the greatest average difference, but well within the interval of the standard deviation.

- Small AUC size and $AUC_{VS} \gg AUC_{SAS}$.
  - Small AUC is chosen with AUC_{SAS} at approximately 50 % of the average AUC_{VS} size. 173.29 Pa s is chosen as this small value. None of the iNPH patients and two of the sNPH patients had a smaller average AUC_{VS} value than this.
  - The size difference between the AUC values is chosen at 5 %, as in the previous point.

It should be noted that as the pressure was sampled at 200 Hz, it means we use 200 data points per second in our simulations. In other words, a higher sampling rate would result in a corresponding increase in data points per cardiac cycle.

Note that the average AUC difference for the sNPH patients gives us a transmantle gradient with a higher pressure in SAS.

Normalization

These curves were normalized in the following ways.

- The curve for AUC_{SAS} was set such that its AUC value was zero. In effect this means that the total pressure on the surface during the cardiac cycle was zero.

- The curve for AUC_{VS} was set such that its AUC value was zero *plus* the difference. In effect, this gives a slight pressure pointing away from the center.

7.4 Visualising the results with Paraview

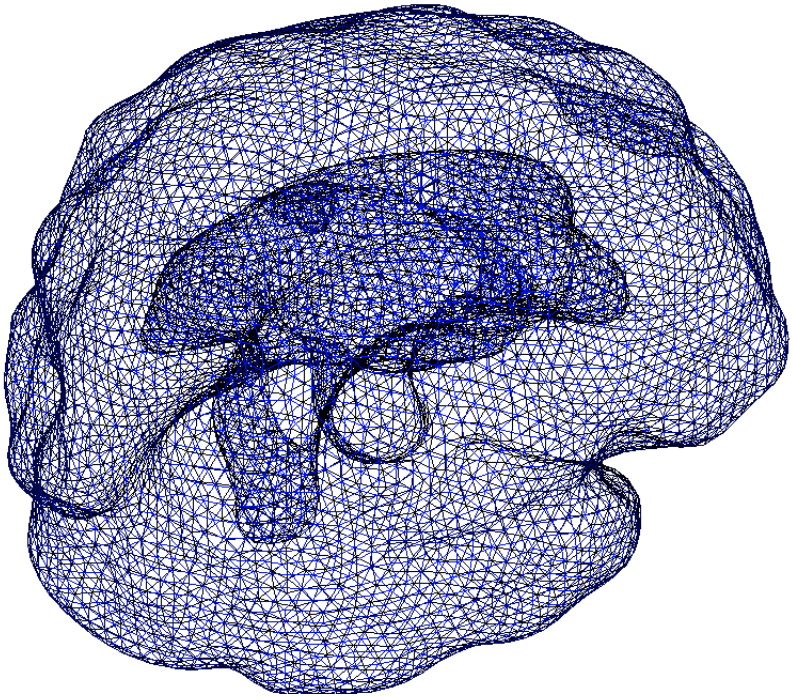
Paraview [17] are used to analyze the solution of simulations. Its `slice` tool is used to extract information about the displacement and von Mises stress on the surfaces. It is also be used to provide a visual of the stress distribution of the inside of the models.

Displacement

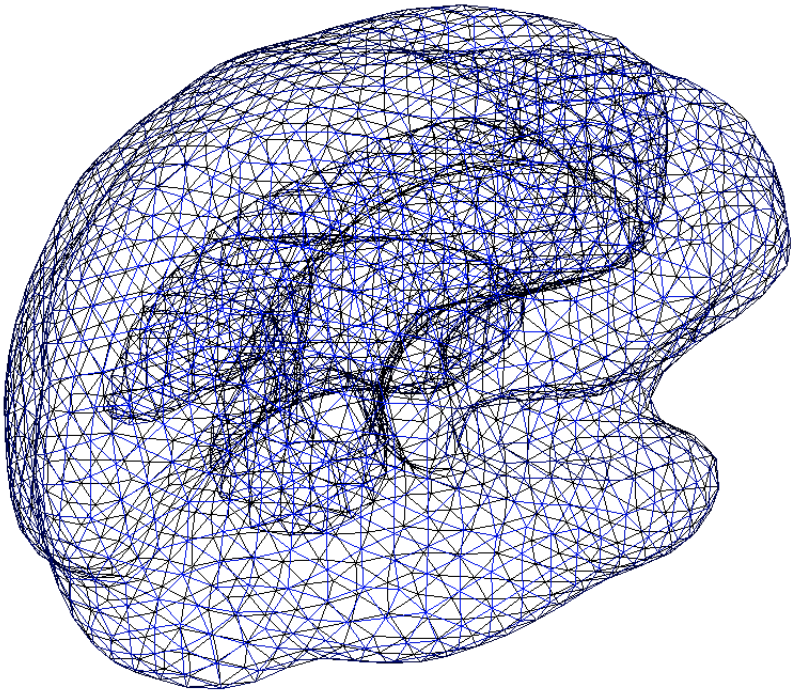
The displacement, as defined by the distance from the original position to the new position, can be displayed in Paraview. In the cases where there is only one time-step, i.e. one time-step, the displacement vector u is derived from the difference between the starting position and the time-step. In more general terms, for more time-steps, the displacement is calculated as the difference between the position at time-step n and time-step $n + 1$. This is one of the standard display settings in Paraview. In both the spherical model and mesh of the brain, slices are taken from the spheres or brain mesh to view cross-sections of the displacement. Where there are several time-steps, i.e. in the simulation of a cardiac cycle, we save data from each time-step. The displacement viewed is the displacement between time-step n and time-step $n + 1$. This is, however, only good for visual purposes. When calculating the displacement from time step zero to the final time step, we calculate the distance the mesh moves from its starting position.

For the spherical model, the displacement will be close to symmetrically distributed. The symmetrical distribution is due to the symmetric nature of the model. However, for the brain mesh, which is not symmetric, the displacement will be more varied. In order to give a more thorough understanding of the displacement, the results for the brain model will partly be presented in images of cross-sections of the brain.

The von Mises stress will be extracted in the same manner as the displacement.



(a) The most detailed brain mesh, seen from above and to the side.



(b) A less detailed brain mesh, seen from above and to the side.

Figure 7.2: Two brain meshes. We ended up using the least detailed one in the simulations, due to the computational resources needed for the most detailed brain.

Chapter 8

Results

The results are divided into three stages. They are divided in order to make a clear presentation of how the results tie together, and how they are obtained. The second stage therefore builds on stage I, while the third stage is based on stages I and II. All parameters collected from the literature are used in our simulations, except the parameters from Dutta-Roy et al. [10] where the brain is modeled as an incompressible material. That the brain is incompressible is not an unrealistic assumption, but it leads to very unstable simulations. For a quick review of the parameters, see Table 7.1.

Stage I The first results are from the simulation of a sphere inside a sphere. The inner sphere represents the ventricles, while the outer sphere is the outside of the brain tissue, with the subarachnoid space and skull outside. Non-iterative simulations on the sphere give us an idea of what kind of displacement and von Mises stress to expect from the complete brain model. These results are compared with an analytical solution. This comparison also involves considering the simulated results when the mesh is refined.

Stage II The spherical model is now treated as the brain model. Curves representing both typical and non-typical cardiac cycles are used to simulate the pressure gradient during a cardiac cycle. The displacement and von Mises stress is calculated. Both Robin and mixed boundary conditions are used.

Stage III The brain mesh created in Chapter 3 is now used instead of the spheres. Non-iterative simulations are used to ascertain the expected displacement and von Mises stress on the structure. These are compared to the results from the spherical model. Then typical and non-typical curves representing cardiac cycles are used to estimate the displacement and von Mises stress after a cardiac cycle. The model used has two different sets of boundary values. The first model will have one Dirichlet boundary and one Neumann boundary. The second will have two small

Dirichlet boundaries, and one Neumann boundary. The pressure curves, see Chapter 7.3, representing cardiac cycles at VS and SAS are used to define the pressure on the Neumann boundary.

8.1 Stage I

To gain an idea of how great displacements and von Mises stresses one can expect, simulations are made on the sphere. A Neumann boundary is placed on the inner sphere. The pressure on the Neumann boundary is set to 2 Pa and 20 Pa. A Dirchlet condition is applied to the outer sphere, thereby locking it in place. The results are displayed in Table 8.1 and Figure 8.1. The figure shows two-dimensional slices of the symmetric three-dimensional mesh.

We have an analytical solution to a three-dimensional problem, translated into a one-dimensional linear elasticity problem by way of spherical coordinates and a few assumptions. The solution is provided in the lecture notes from the course ME211 at the University of Oslo [20]. This analytical solution assumes Neumann boundaries on the whole boundary. However, pressure is only applied on one of the boundaries. This is not an exact comparison, but it will help provide guidance for our solution values.

The analytical solution to the one-dimensional problem is

$$u = \frac{r}{E}((1-\nu)\sigma_{\phi\phi} - \nu\sigma_{rr}) \tag{8.1}$$

where

$$\epsilon_{\phi\phi} = -\frac{p_g a^3(2r^3 + b^3)}{2r^3(a^3 - b^3)} \tag{8.2}$$

and E is Young's modulus, ν is Poisson's ratio, p_g is the pressure on the inner boundary a and b is the outer boundary where the pressure is set to zero.

Note that the analytical solution for the one-dimensional problem is approximately a factor of ten away from our computed solution. We apply a Robin condition with pressure set to 20 Pa, and $k = 10^6$, with a Neumann boundary with pressure set to zero on the outer surface. It gives essentially the same results, with two differences. First, the maximum displacements are slightly larger, approximately 10% increase. The only exception to this is a 50% increase in displacement, which is for the second set of parameters by Dutta-Roy et al. [10]. Their parameters are different in that the Poisson's ratio is set to 0.49. This translates to an almost incompressible medium.

8.2 Stage II

As seen in the previous section, several of the elasticity parameters are very similiar and therefore provide very similar results. One of the parameter sets simulates an incompressible medium, which we do not simulate, as this leads to

Source	Max Displacement Pressure at 20 Pa	Max displacement Pressure at 2 Pa	Analytical solution 20 Pa	Analytical solution 2 Pa
Dutta-Roy (a) et al. [10]	NaN	NaN	NaN	NaN
Dutta-Roy (b) et al. [10]	0.324	0.032	0.012	0.001
Dutta-Roy (c) et al. [10]	0.594	0.059	0.0868	0.009
Humphrey & Smith [33]	0.050	0.004	0.0044	0.0004
Taylor & Miller [38]	0.427	0.043	0.054	0.005
Wirth & Sobey [35]	0.428	0.043	0.052	0.005
OCCAM Fields MITACS [6]	0.012	0.001	0.002	0.0002

Table 8.1: Displacement after one time-step of a pressure of 2 Pa and 20 Pa applied to the inner sphere. The results from our numerical simulations are displayed in the first two columns. The last two columns are the analytical solution for the one-dimensional solution. The displacements are estimated in mm.

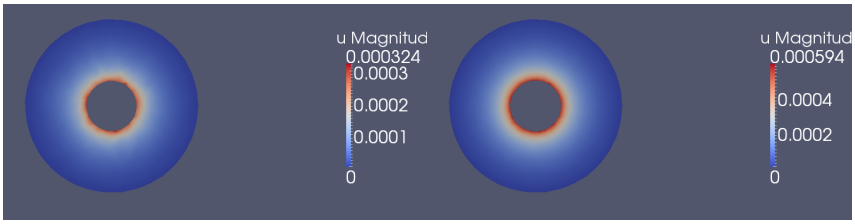
locking. So in order to simplify the overview of results we only run simulations on four of the sets of parameters.

- Dutta Roy et al. (b) [10]. A nearly incompressible brain with Poisson's ratio equal 0.49 and Young's modulus set at 464.19 Pa.
- Humphrey and Smith [33] uses a Poisson's ratio of 0.4 and Young's modulus of 5000 Pa.
- Taylor and Miller [35] refers a Poisson's ratio of 0.35 and Young's modulus of 584.4 Pa.
- The proceeding from OCCAM-Fields-MITACS Biomedical Problem Solving Workshop, 2009 [6] used a shear modulus, μ, of 8000 Pa and Poisson's ratio of 0.35.

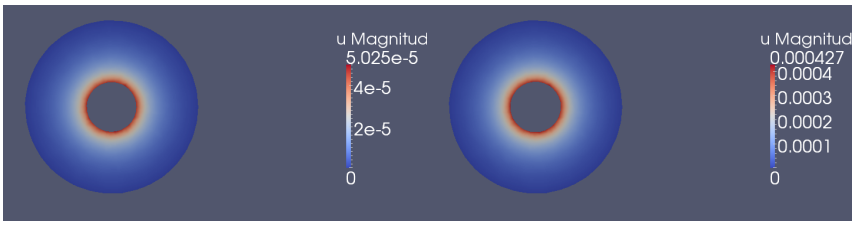
We run simulations on the spherical model where the outer sphere is locked in place with a Dirichlet condition, i.e., no displacement. The inner boundary is a Neumann boundary. The pressure on the Neumann boundary is a curve defined as the difference between the pressure in the VS and the pressure in SAS.

The results can be seen in Table 8.2.

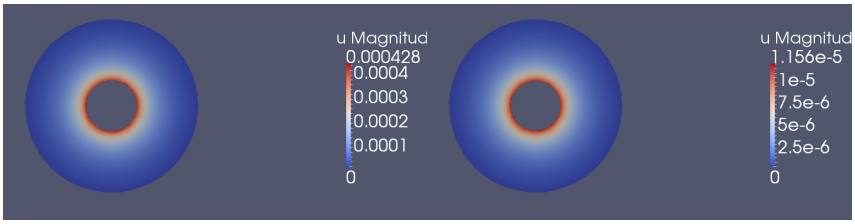
While the deformations for one time-step are fairly small, they can get quite large when there are enough time-steps. As seen in Table 8.2 there are some large displacements. Especially for the parameter values given by



(a) Dutta-Roy (b) (b) Dutta-Roy (c)



(c) Smith & Humphrey (d) Taylor & Miller



(e) Wirth & Sobey (f) OCCAM

Figure 8.1: Slices of the displacement for pressure set to 20 Pa on the inner sphere. Note that the displacements are estimated in meter.

Taylor & Miller [38] and Dutta-Roy et al. (c) [10]. When there is a 5 cm distance between the inner and outer sphere, a displacement of nearly 3.3 cm is unrealistic. Especially if one considers that sometimes the displacements move towards the center of the spherical mesh. In some cases this means that the total span of the deformations is up towards 5 cm.

As these displacement values are too large to be realistic, we will only use Humphrey & Smith [33] and the proceeding from OCCAM [6] for the simulations of cardiac cycles on the brain.

Note that the average difference for sNPH has a negative transmantle gradient. This means that the displacement for those parameters is a contraction of the inner sphere.

Cardiac Cycle	Displacement for Dutta-Roy (b) et al. [10]	Displacement for Humphrey & Smith [33]	Displacement for Taylor & Miller [38]	Displacement for OCCAM Fields MITACS [6]
Average AUC Average difference for iNPH	6.4872	0.4559	NaN	0.0987
Average AUC Average difference for sNPH	10.8491	1.7343	14.8780	0.3099
Large AUC $AUC_{VS} >> AUC_{SAS}$	33.3562	14.9263	NaN	3.2324
Small AUC $AUC_{VS} >> AUC_{SAS}$	1.3399	0.3086	0.8961	0.0735

Table 8.2: Maximum change in the radius of the inner sphere after a cardiac cycle. The pressure is the pressure i VS minus the pressure in SAS. The displacement is measured in millimeter. Note that the change for the sNPH curves is for a contraction of the inner sphere.

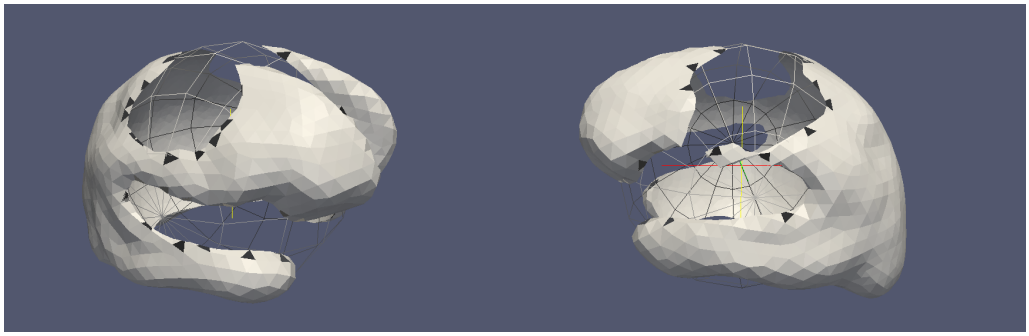
8.3 Stage III

We use two different sets of boundary conditions in this section. Both sets are mixed boundary conditions, with the pressure on the Neumann boundary set to the VS curve minus the SAS curve.

The first set is very similar to the conditions used in Section 8.2. A sphere of radius 55 mm is placed around the center of the brain mesh. The boundary outside of this sphere is given a Dirichlet boundary condition, while the surface inside the sphere is given a Neumann boundary condition. Pressure is applied to the Neumann boundary, while the Dirichlet boundary is locked at zero displacement. The Dirichlet boundary can be considered to represent how the pia mater is connected to the arachnoid membrane through thin strands [37]. These boundary conditions are displayed in Figure 8.2 a-d).

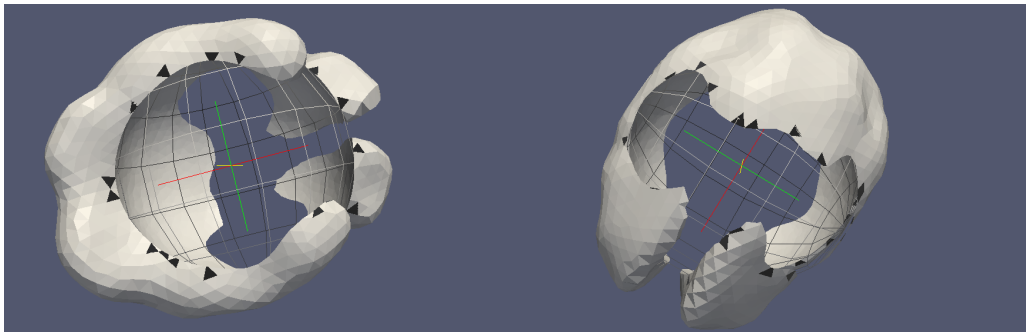
In the second set of boundary conditions we apply two small Dirichlet boundaries to the tip of the brain. This serves to lock the mesh in place, and ensure that we get a solution. We then apply a Neumann boundary condition to the remaining surface. The Dirichlet boundaries can now be considered to represent that the front of the brain is locked in place by the skull, while the rest of the brain remains free to be displaced by the CSF pressure. This is displayed in Figure 8.2 e-f). This second set of boundary conditions will provide an example of how alternative boundary conditions might influence the results.

First we run the simulations on the brain mesh with just one time-step. The results of these simulations can be seen in Tables 8.3 and 8.4. The values have the same order of magnitude as from the simulations on the spherical mesh, see Table 8.1. The displacement distribution for one time-step on the brain mesh is displayed in Figure 8.3, with the first boundary conditions.



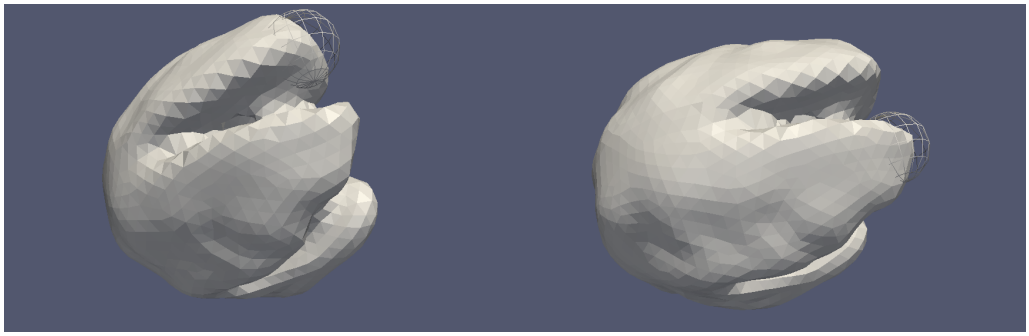
(a) The sphere defining the Dirchlet and Neumann boundary values for the first set of boundary values. Seen from above and to the right.

(b) The sphere defining the Dirchlet and Neumann boundary values for the first set of boundary values. Seen from the left side.



(c) The sphere defining the Dirchlet and Neumann boundary values for the first set of boundary values. Seen from below.

(d) The sphere defining the Dirchlet and Neumann boundary values for the first set of boundary values. Seen from above.



(e) The Dirichlet boundary condition located at the tip of the right brain hemisphere.

(f) The Dirichlet boundary condition located at the tip of the left brain hemisphere.

Figure 8.2: Illustrations of how the boundary conditions are defined in the simulations in Section 8.3.

Note that one can see the following loose relation between the different displacement values for the first boundary conditions defined:

Source	Max Displacement Pressure at 20 Pa	Average displacement Pressure at 20 Pa	Max Displacement Pressure at 2 Pa	Average displacement Pressure at 2 Pa
Dutta-Roy et al. (a) [10]	NaN	NaN	NaN	NaN
Dutta-Roy et al. (b) [10]	0.2109	0.0555	0.0211	0.0055
Dutta-Roy et al. (c) [10]	1.2259	0.2755	0.1226	0.0276
Humphrey & Smith [33]	0.0819	0.0197	0.0082	0.0020
Taylor & Miller [38]	0.8823	0.1983	0.0882	0.0198
Wirth & Sobey [35]	0.8684	0.1962	0.0868	0.0196
OCCAM Fields MITACS [6]	0.0239	0.0054	0.0024	0.0005

Table 8.3: The displacements after one time-step. Average displacement and max displacement, for both 20 Pa and 2 Pa. The displacement is measured in mm. The boundary conditions used are one Dirichlet boundary on the surface outside of a sphere, while the Neumann boundary is on the surface outside of the sphere.

$$\text{Max}(u)_{Brain} \;=\; 2 \cdot \text{Max}(u)_{Sphere} \;=\; 2 \cdot \text{Average}(u)_{Brain} \tag{8.3}$$

When comparing the results, we note that the maximum displacements are roughly twice as great for the second definiton as displacements for the first boundary values. Note also that the average displacement has a greater variation in the difference between the two boundary conditions. The smallest differences in the average displacements are for the parameter values, those used by Humphrey & Smith [33] and Dutta-Roy et al. (b) [10], where Poisson's ratio is highest. In our cases the values are 0.4 and 0.49 respectively.

We then run simulations on the brain through an entire cardiac cycle. For these simulations we only use the first boundary conditions, i.e., one Dirichlet boundary and one Neumann boundary. The reason for this is that with the number of time-steps needed to go through a cardiac cycle, the second boundary conditions leads to an unstable system.

In Table 8.5 the results for a cardiac cycle on the brain is displayed.

Visualization

In our visualizations we use the parameter values from Humphrey & Smith [33]. We choose these parameters because they give reasonable results.

When slices of the brain are provided, there are one to three centimeters between the slices.

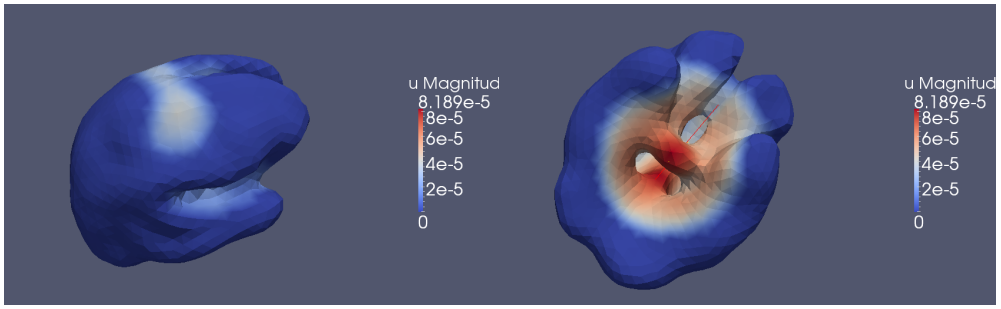
In Figures 8.4, 8.5 and 8.6 we display sagittal, transverse and coronal slices of the brain. Only one time-step is visualized, hence these results coincide with

Source	Max Displacement Pressure at 20 Pa	Average displacement Pressure at 20 Pa	Max Displacement Pressure at 2 Pa	Average displacement Pressure at 2 Pa
Dutta-Roy et al. (a) [10]	NaN	NaN	NaN	NaN
Dutta-Roy et al. (b) [10]	0.1323	0.0765	0.01322	0.0076
Dutta-Roy et al. (c) [10]	2.1137	1.1917	0.2114	0.1192
Humphrey & Smith [33]	0.1195	0.0678	0.0120	0.0068
Taylor & Miller [38]	1.5212	0.8576	0.1521	0.0858
Wirth & Sobey [35]	1.4800	0.8348	0.1480	0.0835
OCCAM Fields MITACS [6]	0.0412	0.0232	0.0041	0.0023

Table 8.4: The displacements after one time-step. Average displacement and max displacement, for both 20 Pa and 2 Pa. The displacement is measured in mm. The boundary conditions used are two small Dirichlet boundaries on the front of the brain, and a Neumann boundary on the remaining surface of the mesh.

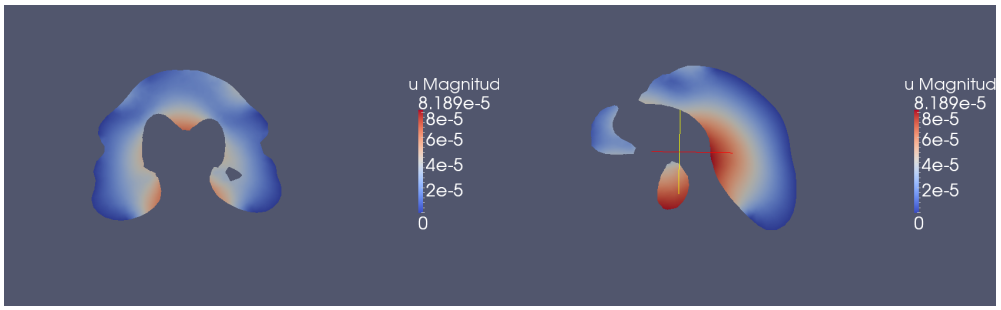
Cardiac Cycle	Max displacement for Humphrey & Smith [33]	Average displacement for Humphrey & Smith[33]	Max displacement for OCCAM Fields MITACS [6]	Average displacement for OCCAM Fields MITACS [6]
Average AUC Average difference for iNPH	0.4998	0.0748	0.0400	0.0055
Average AUC Average difference for sNPH	1.0077	0.1268	0.0744	0.0095
Large AUC $AUC_{VS} >> AUC_{SAS}$	0.5662	0.0758	0.0421	0.0057
Small AUC $AUC_{VS} >> AUC_{SAS}$	0.0550	0.0075	0.0040	0.0006

Table 8.5: Displacement after a cardiac cycle. The pressure is the pressure i VS minus the pressure in SAS. The displacement is measured in millimeter. The boundary conditions used are one Dirichlet boundary outside a sphere around the center, and one Neumann boundary inside of the same sphere.



(a) Displacements on the mesh seen from above.

(b) Displacements on the mesh seen from below.



(c) Displacements on a slice of brain mesh, seen from in front.

(d) Displacements on a slice of brain mesh, seen from the side.

Figure 8.3: The displacements on the brain after one time-step, pressure on the inside set to 20 Pa. The boundary conditions used are one Dirichlet and one Neumann boundary. The elasticity parameters are the ones from Humphrey & Smith [33].

the ones displayed in Table 8.3. The boundary conditions used are the first ones defined, with one Dirichlet boundary and one Neumann boundary with 20 Pa on the surface. From the sagittal and transverse slices we can see that the greatest deformations are centered around the ventricles. This is hardly surprising, considering the boundary conditions. If one also considers the coronal slices, the displacements are also greater at the back of the brain than at the front.

In Figures 8.7, 8.8 and 8.9 we display the sagittal, transverse and coronal slices of the brain when the second boundary values are used, i.e., two small Dirichlet boundaries and one Neumann boundary. Only one time-step is visualized, hence the results coincide with the ones provided in Table 8.4.

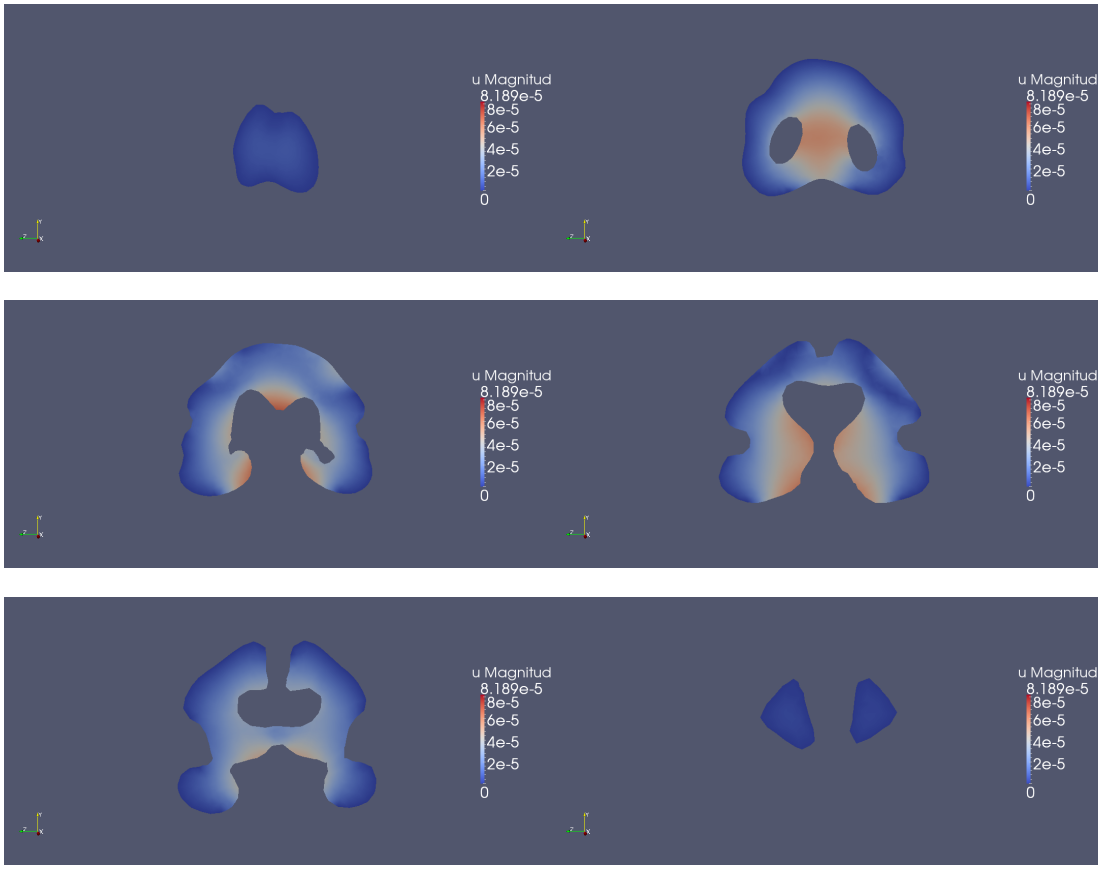


Figure 8.4: **Sagittal slices** of the displacement on the brain for pressure set to 20 Pa on the inside. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

8.4 von Mises stress

In order to consider whether plastic deformations may occur, we consider the von Mises stress, discussed in Chapter 6. We first consider the case for one time-step on the spherical model, with 20 Pa of pressure applied to the Neumann boundary on the inner sphere.

The von Mises stress has its greatest values on very few elements. Most of the mesh has small values. The mesh elements with the highest von Mises stress might be caused by the mesh structure. However, it also coincides with where the greatest deformations are located, see Figure 8.1. The distribution of the von Mises stress is displayed in Figures 8.10.

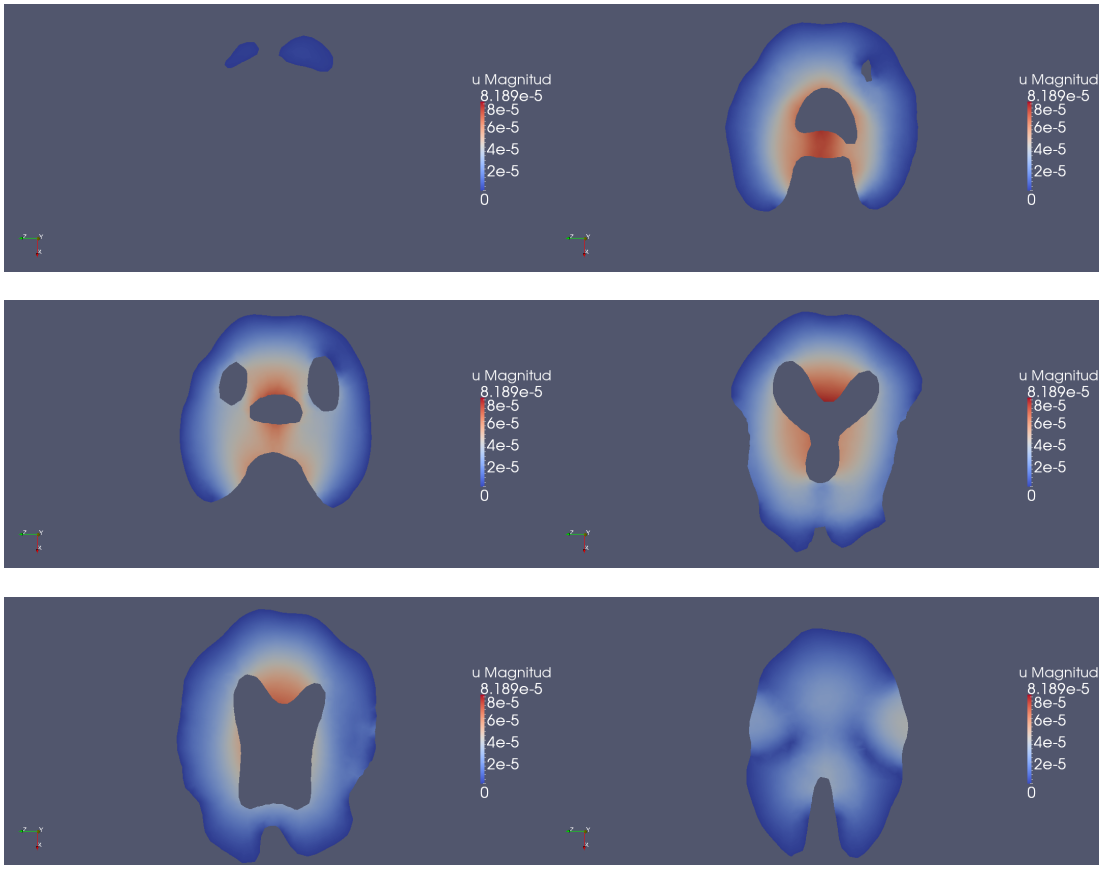


Figure 8.5: **Transverse slices** of the displacement on the brain for pressure set to 20 Pa on the inside. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

When considering the brain mesh, see Figure 8.11. The highest areas of stress coincides with where the displacement is high, see Figure 8.3, and the mesh is curved. A more refined mesh might give lower stress values in these areas, though that is highly speculative. The boundary conditions used here are the first defined, with one Dirichlet and one Neumann boundary.

As with the displacements of the brain, we consider sagittal, transverse and coronal slices of the brain after one time-step of 20 Pa pressure applied to the Neumann boundary. In Figures 8.12, 8.13 and 8.14 we consider the von Mises stress when one Dirichlet and one Neumann boundary is applied in the simulations. Again, note that the stress is highest where the displacement values are high, and the mesh is curved.

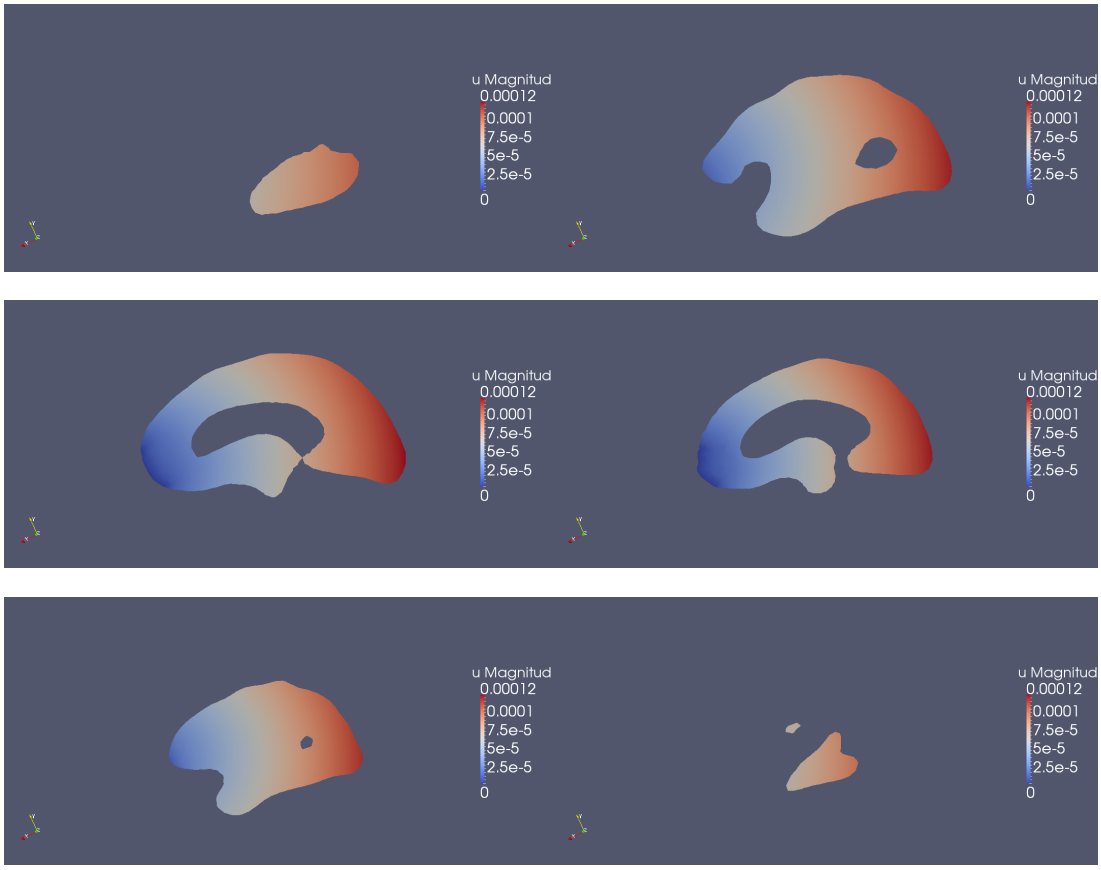


Figure 8.6: **Coronal slices** of the displacement on the brain for pressure set to 20 Pa on the inside. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

In Figures 8.15, 8.16 and 8.17 we display sagittal, transverse and coronal slices of the von Mises stress when we apply two small Dirichlet boundaries and one Neumann boundary to the brain mesh. We now see, particularly in the coronal slices in Figure 8.17, that the von Mises stress is particularly large where the Dirichlet boundaries meet the Neumann boundary. As this is where the fixed surface meets the flexible surface, this is not very surprising.

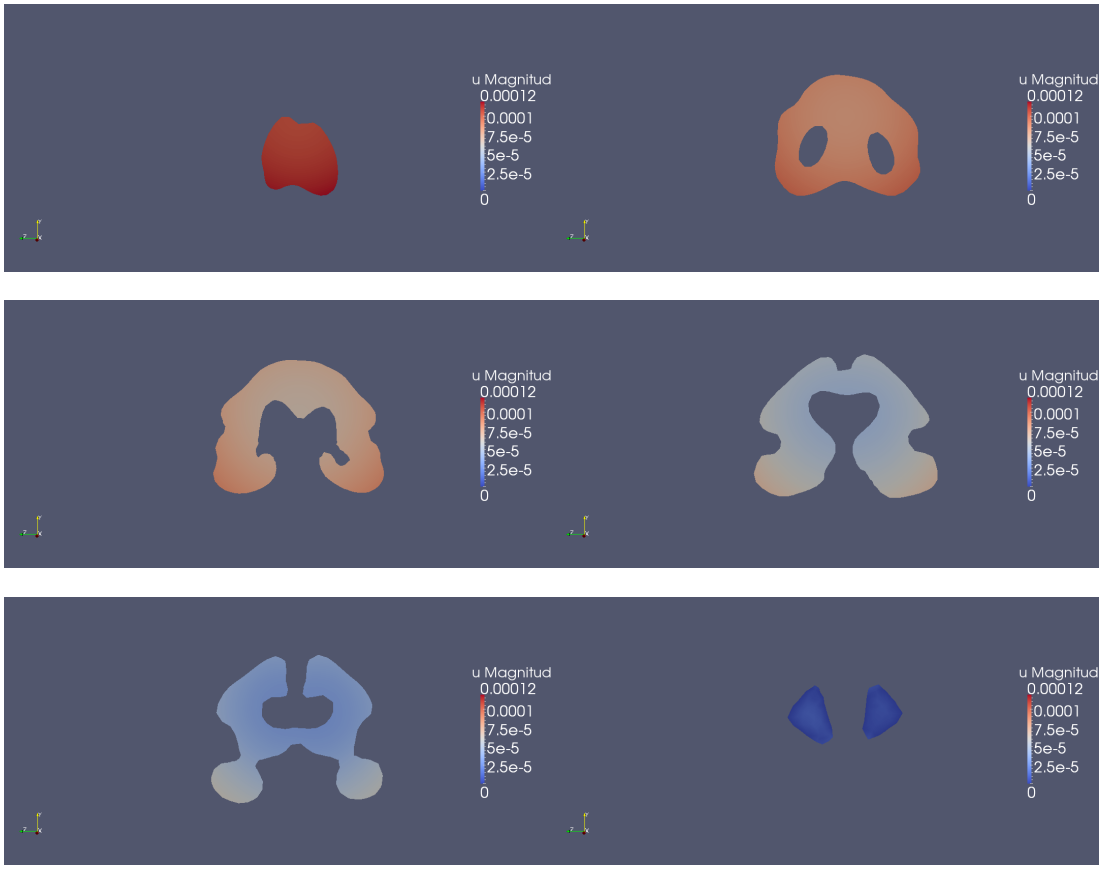


Figure 8.7: **Sagittal slices** of the displacement on the brain, with the pressure set to 20 Pa on the entire surface, except for the front of the brain. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

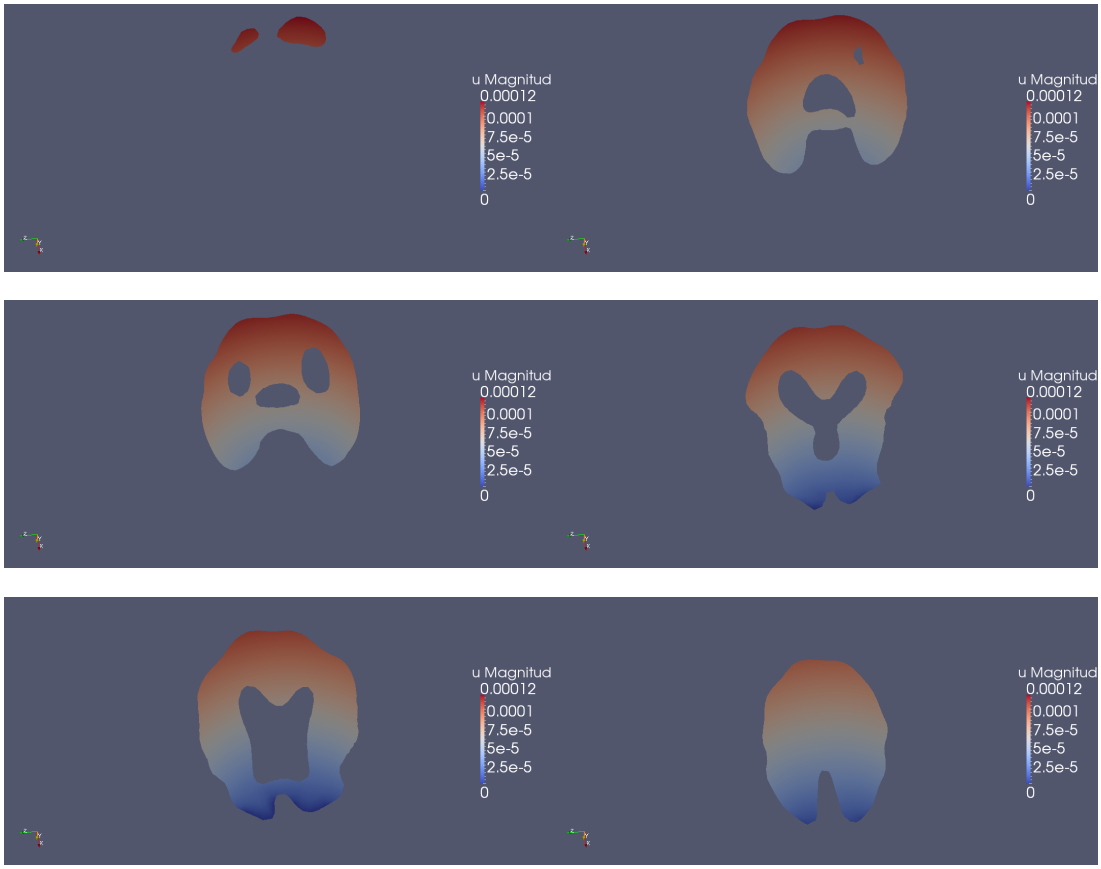


Figure 8.8: **Transverse slices** of the displacement on the brain, with the pressure set to 20 Pa on the entire surface, except for the front of the brain. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

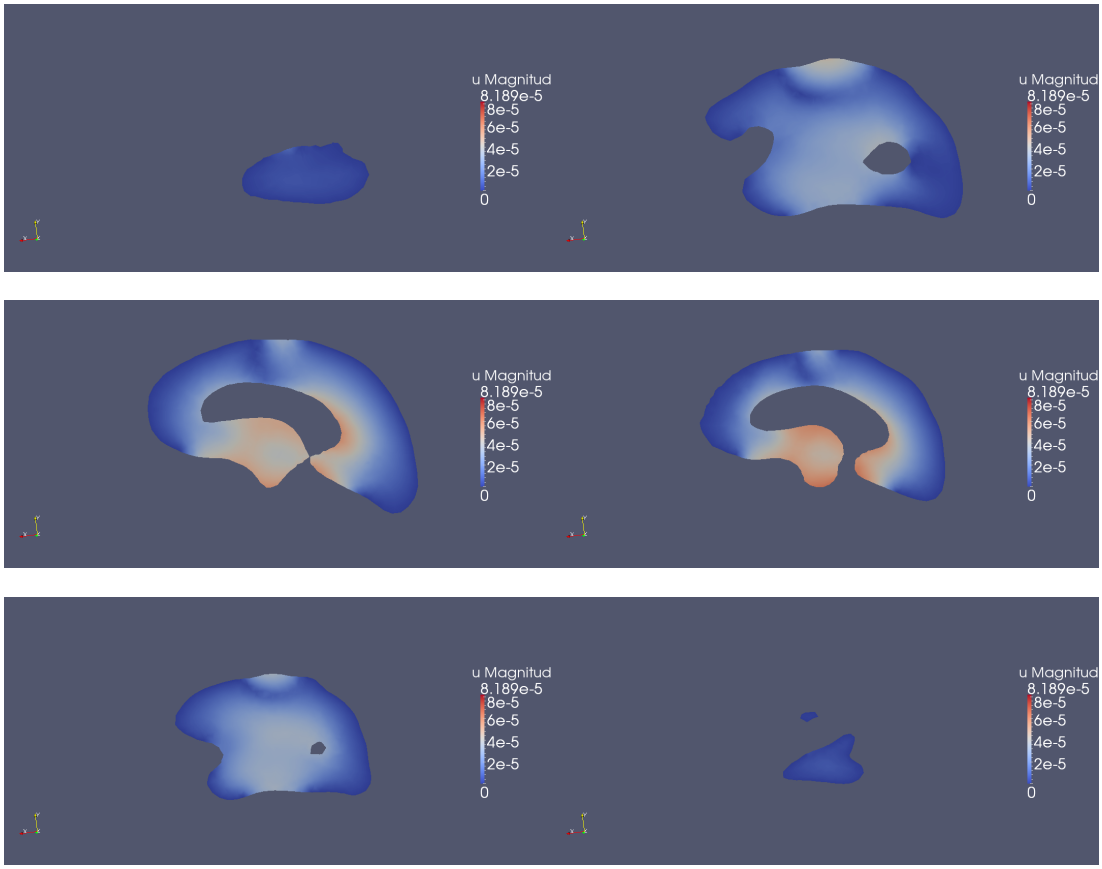
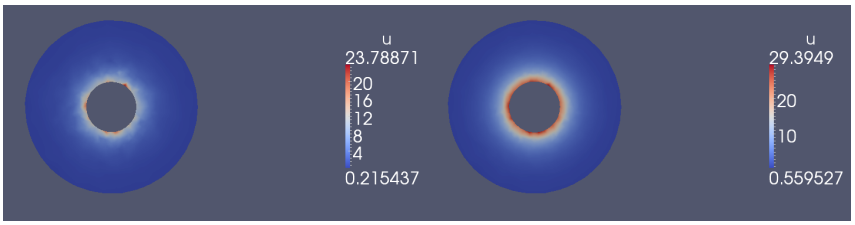


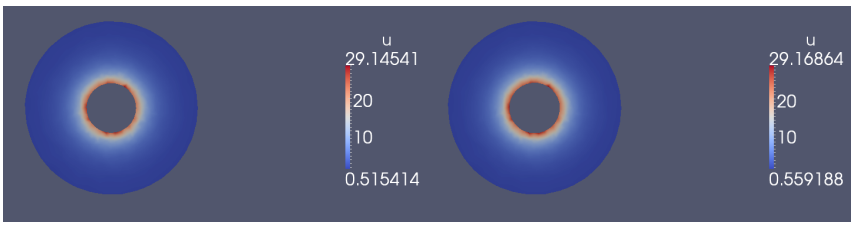
Figure 8.9: **Coronal slices** of the displacement on the brain, with the pressure set to 20 Pa on the entire surface, except for the front of the brain. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

Source	Max von Mises stress Sphere	Max von Mises stress Brain
Dutta-Roy et al. (a) [10]	NaN	NaN
Dutta-Roy et al. (b) [10]	36.48	9.44
Dutta-Roy et al. (c) [10]	28.29	49.14
Humphrey & Smith [33]	35.25	41.73
Taylor & Miller [38]	36.07	49.98
Wirth & Sobey [35]	36.08	49.32
OCCAM Fields MITACS [6]	35.13	52.14

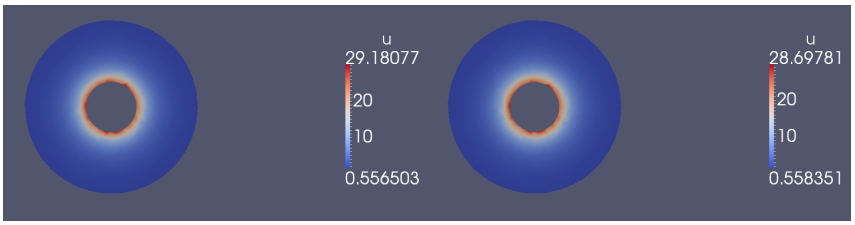
Table 8.6: The von Mises stress for one time-step, spherical and brain mesh, with 20 Pa applied to the Neumann boundary.



(a) Dutta-Roy (b) (b) Dutta-Roy (c)

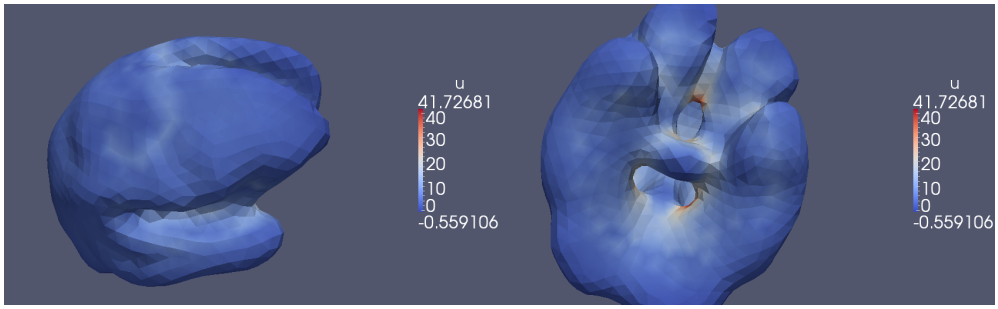


(c) Smith & Humphrey (d) Taylor & Miller

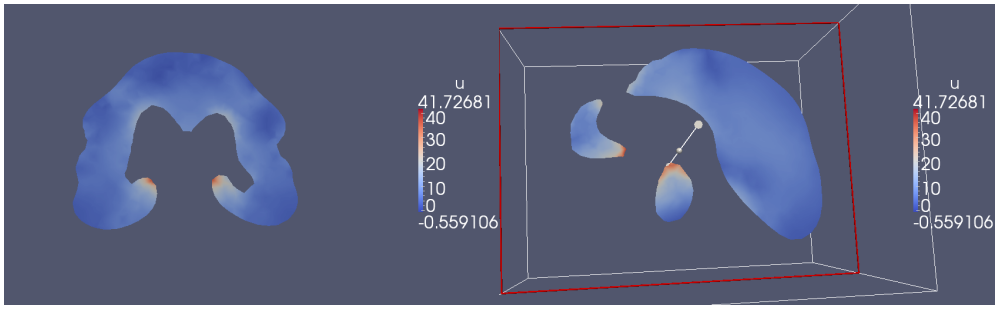


(e) Wirth & Sobey (f) OCCAM

Figure 8.10: Slices of the von Mises stress for pressure set to 20 Pa on the outer sphere.



(a) Brain mesh seen from above. (b) Brain mesh seen from below.



(c) Slice of brain mesh, seen from in front. (d) Slice of brain mesh, seen from the side.

Figure 8.11: The von Mises stress on the brain after one time-step, pressure on the inside set to 20 Pa.

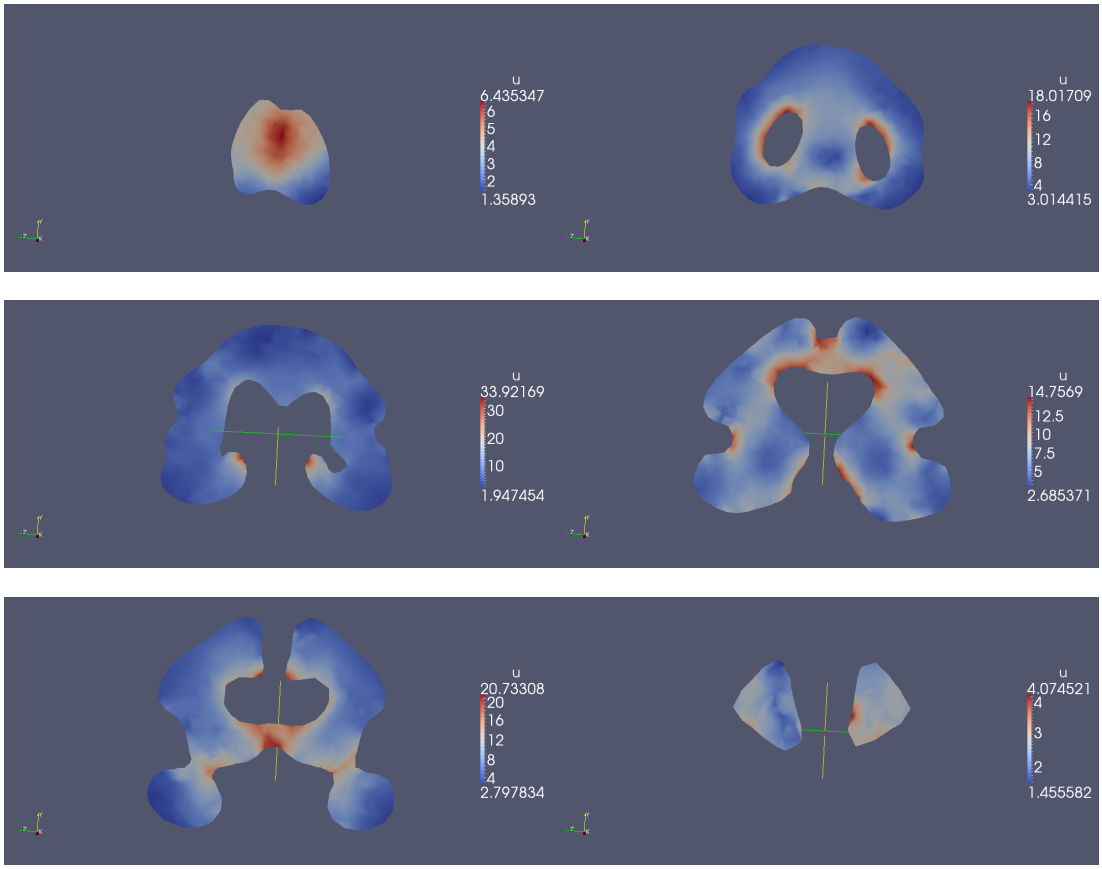


Figure 8.12: **Sagittal slices** of the von Mises stress on the brain for pressure set to 20 Pa on the inside of a sphere. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

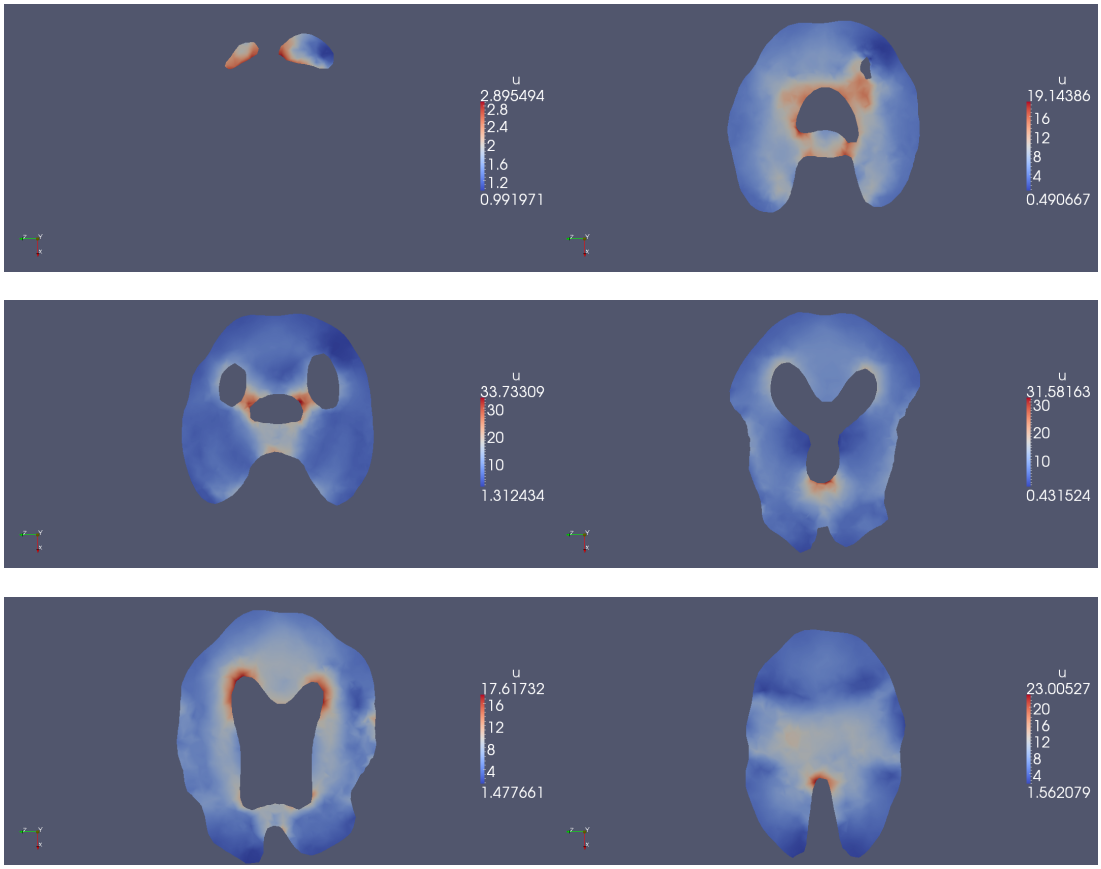


Figure 8.13: **Transverse slices** of the von Mises stress on the brain for pressure set to 20 Pa on the inside of a sphere. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

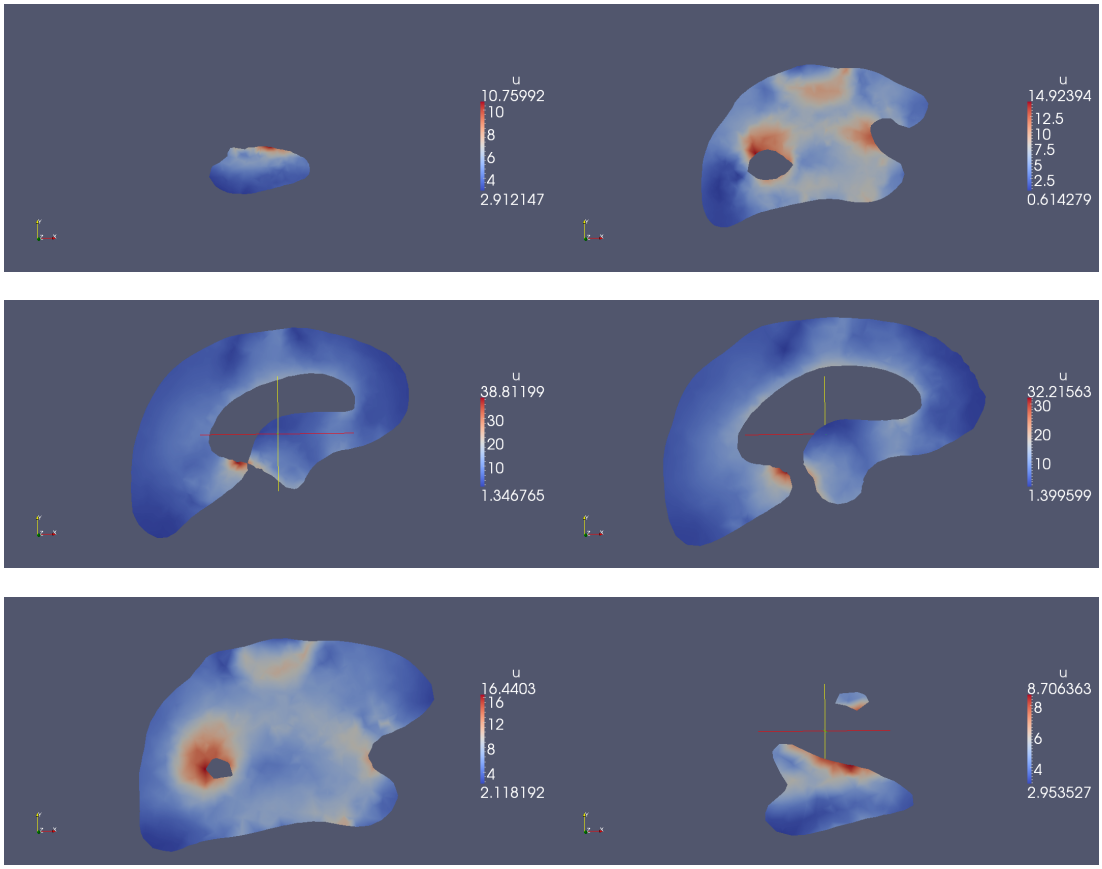


Figure 8.14: **Coronal slices** of the von Mises stress on the brain for pressure set to 20 Pa on the inside of a sphere. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

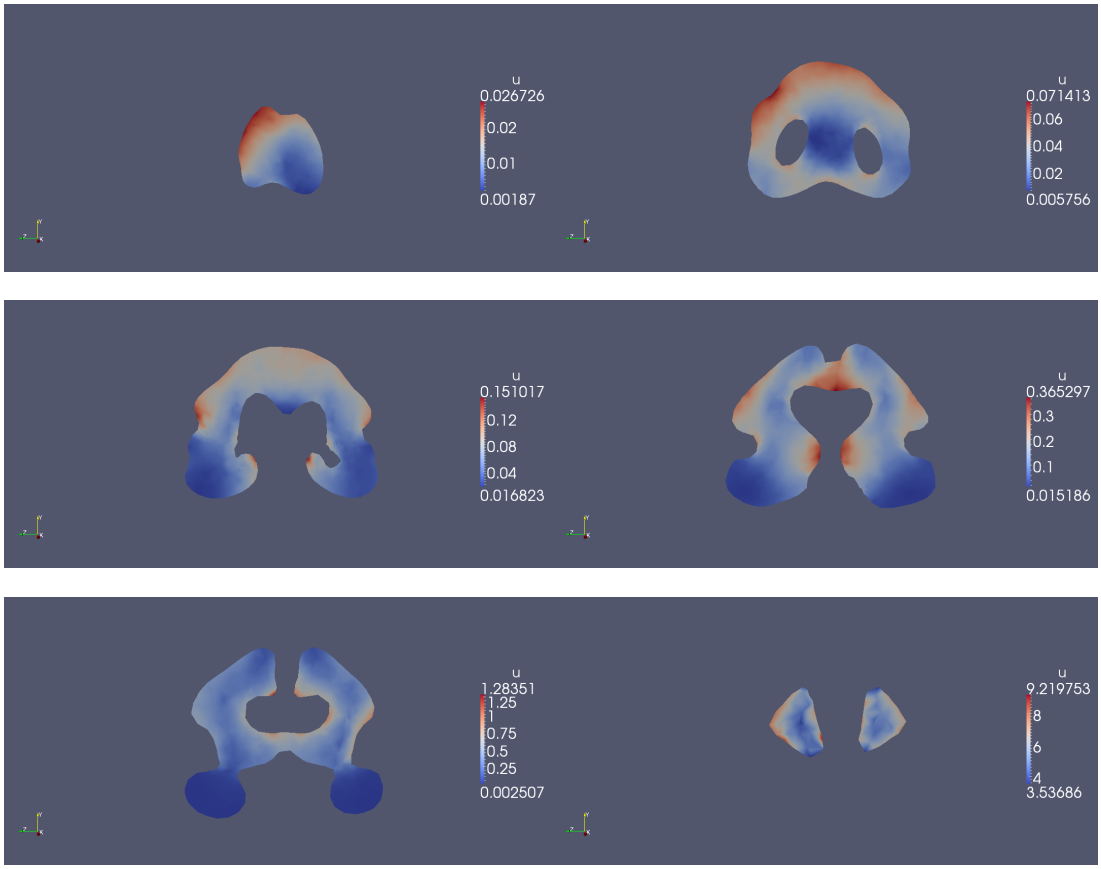


Figure 8.15: **Sagittal slices** of the von Mises stress on the brain, with the pressure set to 20 Pa on the entire surface, except for the front of the brain. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

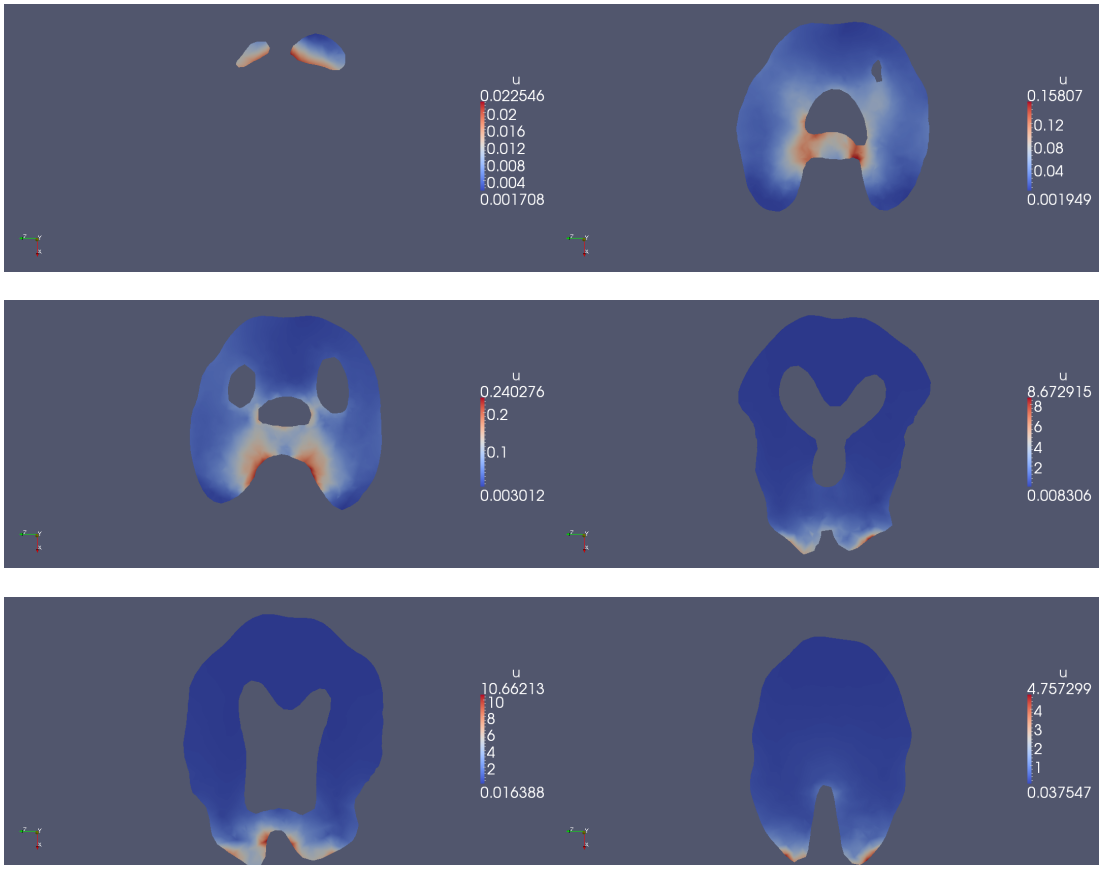


Figure 8.16: **Transverse slices** of the von Mises stress on the brain, with the pressure set to 20 Pa on the entire surface, except for the front of the brain. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

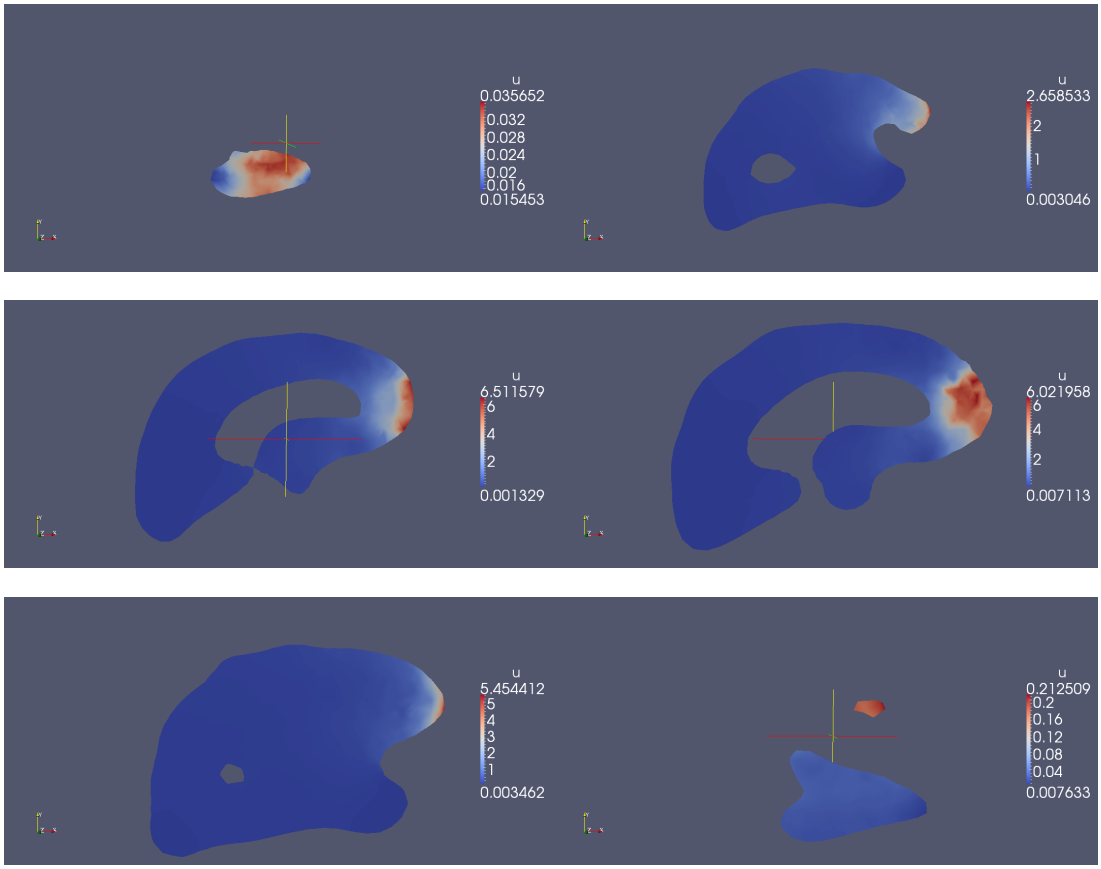


Figure 8.17: **Coronal slices** of the von Mises stress on the brain, with the pressure set to 20 Pa on the entire surface, except for the front of the brain. The elasticity parameters used are from Humphrey & Smith [33], which has the Lamé parameters $\lambda = 7142.86\,\text{Pa}$ and $\mu = 1785.71\,\text{Pa}$. These are the second highest values used, and some of the parameters that gave realistic results with regards to the sizes of the displacements.

Chapter 9

Discussion

It is clear that for our model that certain elasticity parameters discussed in the literature are inaccurate. The elasticity parameters are generally too low for our model to make a realistic approximation of normal and expected scenarios.

It is interesting to note that some of the highest displacements came from the sNPH curve, which has a lower AUC than most of the other curves. However, it does have a notably higher pressure in the SAS than in the VS, leading to contractions instead of expansions of the ventricles.

In general, the higher AUC values leads to greater deformations. This is expected, as a 5% difference in a higher AUC will give a greater transmantle gradient in absolute value than a 5% difference in a lower AUC.

Some of the displacement values for the brain mesh are greater than for the sphere. We suspect that this is because parts of the brain mesh have a greater mass-to-surface ratio locally than the spherical mesh. This would naturally cause greater displacements, as more pressure is exerted on a smaller area.

In our simulations for the displacements of the spherical mesh, see Table 8.1, it is interesting to note the differences we get when we use Robin and Neumann boundaries. The displacements are essentially the same, unless one considers the brain as an incompressible medium, such as Dutta-Roy et al. [10].

An obvious consideration must be the number of time-steps per second. With the use of 200 time-steps a second it is obvious that rather large deformations may occur during a fairly short time.

It is also reasonable to say that the deformations go more than one way. There are substantial differences in the size and direction of the transmantle gradient. Assuming that there is a steady transmantle gradient in either direction is a considerable simplification, and further investigations could look into the possibility of a varying transmantle gradient. There is a possibility that the transmantle gradient in one direction has a greater effect than the transmantle gradient in the other direction.

There is also the possibility of plastic deformations. As seen clearly in

Figure 8.17, the von Mises stress is particularly high in the area where the Dirichlet boundary meets the Neumann boundary. This indicates that the brain is especially vulnerable for plastic deformations where the brain is locally connected to parts of the brain or otherwise locked in place.

9.1 Mesh generation

Generating the mesh proved to be the most problematic part of the preliminary work. The MRI images of the brain scans were not consistent with pixel values. The same pixel value could correspond to both brain tissue and CSF, depending on the location in the brain. This meant that in some cases we had to make a decision whether to count parts of the image as CSF or brain tissue. This was problematic with the grey brain matter, not the white brain matter.

In order to have accurate simulations we required a good approximation of the brain structure. Creating a complex surface mesh was not a problem. However, creating a volume mesh in VMTK required a less complex geometric structure. Particularly the smaller sulci on the outer side of the brain had to be smoothed out considerably. The end result was a rougher representation of the brain, with only the largest structures evident. As the main concern of the simulations of hydrocephalus is the expansion of the ventricles, which are intact in the mesh, the decreased accuracy should not be a problem. We can not, however, rule out that the simplification of the brain might influence the results in some way. Particularly the von Mises stress might have been smaller if the mesh were more refined in the areas where the von Mises stress were high.

A more refined mesh would provide a more nuanced view of the displacements. However, this would also demand considerably greater computational resources.

It is also worth noting that our mesh MRI images were obtained from NPH patients. Therefore the ventricles are already somewhat enlarged.

9.2 Data Analysis

The analysis of the raw pressure data received from Rikshospitalet was the first step of the process. The very first thing done was to observe the shape of the pressure cycles corresponding to cardiac cycles. Then these cycles were identified. The individual and average values were obtained for level of synchronization between measures in CSF and EPI, amplitude, phase-time and area under curve (AUC). This was a fairly straightforward process. The pressure data also contained noise. In order to exclude the large-scale noise, the most extreme cardiac cycles in terms of AUC size were excluded. One weakness is that the pressure data is from patients who already possess enlargement

of the ventricles. Thus we lack data from when the enlargement is occuring. Instead we try to simulate deformations based on how the CSF pressure is *after* the deformations have occured.

Cardiac cycles

We choose to extract cardiac cycles from patients with distinct characteristics. Originally the plan was to use cardiac cycles for every single patient. However, when comparing the AUC sizes, AUC differences, amplitude and phase-time, the patients are very similar. Using average curves would lead to very many similar results, possibly leading to a confusing presentation. There is a statistically significant difference between the AUC sizes of iNPH and sNPH patients. Because of this difference we use the two categories of iNPH and sNPH to differentiate between some of the cardiac cycles used.

It should be noted that out of the ten iNPH patients, four had on average a larger AUC_{VS} than AUC_{SAS}. For six patients it was the opposite, with a smaller AUC_{VS} than AUC_{SAS}. Out of only ten patients, it is hard to say anything for certain concerning the expected distribution. The sample number is simply too small.

However, the standard deviation of the AUC values is considerable. This means we have large differences in both directions of the AUC sizes. One highly speculative theory for why this could still lead to an increase in ventricular size is that when the pressure is greater in SAS the brain structure is not affected to a great extent. When the pressure is increased in the ventricles, the ventricles are enlarged. Wirth [37] justified locking the displacement at the skull by strands connecting the pia mater and the arachnoid. Should that hold, a theory for ventricular enlargement is that the ventricular size only is substantially affected when $AUC_{VS} \geq AUC_{SAS}$.

9.3 Boundary Conditions

As discussed in Section 6.7, there are several options for boundary conditions. The ideal implementation for our simulations would be if one could place Neumann boundary conditions on the whole surface. As this is unstable that could not be accomplished. We can also tell directly from Korn's Inequality, Theorem 6.3.1, that a pure traction problem will not work.

Attempts were made to use Robin boundary conditions. A Robin boundary was hoped to give enough stability, yet keep it possible to keep pressure specified on the whole surface. Possibly with mixed boundary conditions with one or more Neumann boundary conditions in addition to the Robin boundary. This was stable for small displacements, but only for a very limited (< 20) number of time-steps. In the end we decided to use mixed boundary conditions. In the end, implement a Neumann boundary condition on the inner surface, and a Dirichlet boundary condition on the outer surface.

Mixed boundary conditions

For the spherical mesh model of the brain a Dirichlet boundary was implemented on the outer boundary. This represents how the outer edges of the brain are connected to the skull. As iNPH is typically characterized by increased ventricular size, we will thus avoid a deformation in the outer edges of the brain. The inner boundary is set with a Neumann boundary condition. The pressure on this boundary was first set to a constant pressure through one time-step. This was done to create an idea of what kind of displacements to expect. This caused some concern of how to implement the two different estimates for pressure. The final model consisted of simply running one simulation with the pressure set to AUC_{VS} minus AUC_{SAS}. This is more thoroughly described in Section 8.2.

The process used on the spherical model was also used on the brain model. The only difference is that the outer surface had to be divided up into smaller boundaries. Parts of the outer surface are also left open, i.e., unconnected to the skull.

Alternative boundary conditions

An alternative was to use a Robin boundary condition with a very small displacement factor. The pressure measures would then have been implemented with the VS measures placed in the ventricles, close to the center of the brain. The SAS pressure would have been set on the outer surface of the brain. The surfaces between these two points (center and outer surface) would have been a linear combination of these two. Possibly a non-linear combination as well, in order to take into account the brain structure. Points close by in linear distance might be far away from each other if one had to travel along the same channels as the CSF.

9.4 Considerations of our model

Limitations

There are several weaknesses in our model that should be taken into consideration.

Linear elasticity is by definition limited to small deformations. The stress-strain relationship is only close to linear when the deformations are small. If the deformations get too large the model will no longer be realistic in modelling hydrocephalus, and plastic deformations may occur.

The model does not take into account how the material properties, i.e. the elasticity parameters, might change during deformations. This could affect the results, as the rate of deformations would most likely change. The size and direction of these changes would probably have *some* effect on the results.

In our segmentation we assume that grey and white brain matter have the same elastic properties. There are conflicting studies regarding this subject.

The sizes of the displacements through a cardiac cycle are very much influenced by the *number of time-steps* we use. We use 200 time-steps a second, as our pressure measures were estimated at 200 Hz. However, as we use a quasi-static linear elasticity the deformations occur at the same time as the pressure is applied. This means we get a change in the pressure 200 times a second, with a new equilibrium of the tissue distribution 200 times a second. Had the pressure been recorded at 300 Hz it is likely we would have seen an approximate increase in our displacement values of 50%.

Our pressure data is obtained from patients diagnosed with iNPH and sNPH. In other words, our pressure data is from *after* the deformations have occured. Trying to simulate how the changes occured based on data from after these changes occured is an obvious weakness in our model.

We do not possess any information regarding how the pressure is during the deformations. It is possible that the variations we have found in the pressures in the VS and the SAS are within the normal threshold for healthy humans. Although the AUC sizes seems to coincide with what has been previously reported with regards to increased CSF pressure pulse in hydrocephalus patients [11, 26].

The pressure on the Neumann boundary is constant over the entire boundary. Clearly this might not be the case, as the pressure is in fact inside the CSF. The CSF flows through the VS and the SAS, as well as the spinal cord. This flow could possibly cause the pressure to be very different depending on the local brain structure.

Strengths of our model

This is, as far as we know, the first study to consider hydrocephalus on a brain mesh, instead of a constructed structure. As such, we have shown that it is possible to obtain more accurate structures for brain simulations. If one compares our numerical solutions to the analytical solutions presented in Table 8.1, one might consider how the dimension influences the simulated results and the choice of elasticity parameters.

We have added to the growing body of studies considering the elasticity parameters of the brain tissue, and, under certain conditions, shown that some of the parameters used in the literature are inaccurate.

Our analysis of the area under the curve of the pressure signals support the previous studies [11, 26] that consider the amplitude of the pulsatile pressure. When considering the substantial standard deviation, and thereby variation, of the differences in the AUC we provide a more complex, and hopefully more complete, picture of the behaviour of the ICP.

While we did not find a statistically significant difference between the AUC differences in iNPH and sNPH, there is a difference in the few patients from

whom we possess pressure data. Further studies could investigate whether any such differences exist.

Error estimation

The analytical solution for the one-dimensional problem provides a check for how great deformations one can expect. As our model is in three dimensions, the analytical solution is not exact. It is possible that we could have used radial coordinates to gain a better model. Our model also includes a Dirichlet boundary condition, as opposed to two Neumann boundary conditions in the analytical solution. It does, however, provide an idea of the magnitude of the analytical solution.

9.5 New fields of investigation

There are several related subjects that could be useful in further studies of hydrocephalus.

Different models of elasticity should be used, e.g., non-linear elasticity or hyperelasticity.

It could be interesting to consider how the flow of the CSF influence the pressure. If, for instance the Navier-Stokes equations, it could be possible to create a map of the velocity of the CSF. The velocity could then be used to estimate the pressure all over the surface of the brain. One could then create more detailed and specific boundary conditions.

This would make it possible to get a more detailed view of how the brain is deformed.

As seen in the results, the size of the displacements is different in different areas of the brain. One further field of study could be to consider the function of the areas with the greatest displacements. A more refined mesh is a natural next step.

Bibliography

[1] Meshlab. `http://meshlab.sourceforge.net/`.

[2] Simplified wrapper and interface generator. `http://www.swig.org/`.

[3] Vmtk - the vascular modeling toolkit. `http://www.vmtk.org/`.

[4] Anthony H. Aletras. Basic mri physics. In Raymond Y. Kwong and Christopher P. Cannon, editors, *Cardiovascular Magnetic Resonance Imaging*, Contemporary Cardiology, pages 1–31. Humana Press, 2008. 10.1007/978-1-59745-306-6_1.

[5] Ivo Babuka and Manil Suri. Locking effects in the finite element approximation of elasticity problems. *Numerische Mathematik*, 62:439–463, 1992.

[6] Ronald Begg and Kathleen Wilkie, editors. *A Mechanism for Ventricular Expansion in Communicating Hydrocephalus*, 2009.

[7] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 2008.

[8] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland Pub. Co, 1978.

[9] Jay L. Devore and Kenneth N. Berk. *Modern Mathematical Statistics With Applications*. Thomson Brooks/Cole, 2007.

[10] Tonmoy Dutta-Roy, Adam Wittek, and Karol Miller. Biomechanical modelling of normal pressure hydrocephalus. *Journal of Biomechanics*, 41:9, 2008.

[11] Per Kristian Eide and Terje Saehle. Is ventriculomegaly in idiopathic normal pressure hydrocephalus associated with a transmantle gradient in pulsatile intracranial pressure? *Acta Neurochirurgica*, 2010.

[12] Per Kristian Eide and Wilhelm Sorteberg. Diagnostic intracranial pressure monitoring and surgical management in idiopathic normal pressure hydrocephalus: A 6-year review of 214 patients. *Neurosurgery*, 66, 2010.

[13] K. Eriksson, D. Estep, P. Hansebo, and C. Johnson. *Computational Differential Equations*. Studentlitteratur, 1996.

[14] Lawrence C. Evans. *Partial Differential Equations, second edition.* American Mathematical Society, 2010.

[15] Kang Feng and Zhong-Ci Shi. *Mathematical Theory of Elastic Structures.* Springer, 2001.

[16] Aristotelis S. Filippidis, M. Yashar S. Kalani, and Harold L. Rekate. Hydrocephalus and aquaporins: lessons learned from the bench. *Child's Nervous System*, 27:27–33, 2011.

[17] Amy Henderson. *The ParaView Guide: A Parallel Visualization Application.* Kitware Inc., 2004.

[18] J. Hoff and R. Barber. Transcerebral mantle pressure in normal pressure hydrocephalus. *Archives of Neurology*, 31(2):101–105, 1974.

[19] Mariusz Kaczmarek, Ravi P. Subramaniam, and Samuel R. Neff. The hydromechanics of hydrocephalus: Steady-state solutions for cylindrical geometry. *Bulletin of Mathematical Biology*, 59:295–323, 1997.

[20] Hans Petter Langtangen. Forelesninger i me211. Lecture Notes, August 2000.

[21] Tom Lindstrm. *Kalkulus.* Universitetsforlaget, 2006.

[22] A. Logg, K.-A. Mardal, and G. N. Wells et al. *Automated Solution of Differential Equations by the Finite Element Method.* Springer, 2012.

[23] Anders Logg and Garth N. Wells. Dolfin: Automated finite element computing. *ACM Trans. Math. Softw.*, 37(2):20:1–20:28, April 2010.

[24] Anders Logg, Garth N. Wells, and Johan Hake. Dolfin: a c++/python finite element library. In Anders Logg, Kent-Andre Mardal, and Garth Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering*, pages 173–225. Springer Berlin Heidelberg, 2012.

[25] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH '87 Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 21, 1987.

[26] T. Matsumoto, H. Nagai, Y. Kasuga, and K. Kamiya. Changes in intracranial pressure (icp) pulse wave following hydrocephalus. *Acta Neurochirurgica*, 82:50–56, 1986.

[27] P. Mildenberger, M. Eichelberg, and E. Martin. Introduction to the dicom standard. *European Radiology*, 12, 2002.

[28] T. Nagashima, N. Tamaki, S. Matsumoto, B Horwitz, and Y. Seguchi. Biomechanics of hydrocephalus: a new theoretical model. *Neurosurgery*, 21, 1987.

[29] M. Nic, J Jirat, and B. Kosata. *IUPAC: Compendium of Chemical Terminology, 2nd edition*. Blackwell Scientific Publications, 1997.

[30] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[31] Harold L. Rekate. A consensus on the classification of hydrocephalus: Its utility in the assessment of abnormalities of cerebrospinal fluid dynamics. *Childs Nerv Syst.*, 10, 2011.

[32] Frederick H. Sklar, Jan T. Diehl, Chester W. Beyer, and W. Kemp Clark. Brain elasticity changes with ventriculomegaly. *Journal of Neurosurgery*, 53(2):173–179, 1980. PMID: 6893601.

[33] Joshua H. Smith and Joseph A.C. Humphrey. Interstitial transport and transvascular fluid exchange during infusion into brain and tumor tissue. *Microvascular research*, 73, 2007.

[34] Hannes Stephensen, Magnus Tisell, and Carsten Wikkels. There is no transmantle pressure gradient in communicating or noncommunicating hydrocephalus. *Neurosurgery*, 50, 2002.

[35] Zeike Taylor and Karol Miller. Reassesment of brain elasticity for analysis of biomechanisms of hydrocephalus. *Journal of Biomechanics*, 37:7, 2004.

[36] B. L. Welch. The generalization of 'student's' problem when several different population variances are involved. *Biometrika*, 34(1/2):pp. 28–35, 1947.

[37] Benedikt Wirth. A mathematical model for hydrocephalus. Master's thesis, St. Anne's College, University of Oxford, 2005.

[38] Benedikt Wirth and Ian Sobey. Analytic solution during an infusion test of the linear unsteady poroelastic equations in a spherically symmetric model of the brain. 2007.

[39] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.